An amortized approach to non-linear mixed-effects modeling based on neural posterior estimation

Jonas Arruda[1], Yannik Schälte[1,2], Clemens Peiter[1], Olga Teplytska[3], Ulrich Jaehde[3], and Jan Hasenauer*[1,2]

[1]University of Bonn, Life and Medical Sciences Institute, 53115 Bonn, Germany
[2]Helmholtz Zentrum München, Computational Health Center, 85764 Neuherberg, Germany
[3]University of Bonn, Pharmaceutical Institute, 53121 Bonn, Germany

August 22, 2023

Abstract

Non-linear mixed-effects models are a powerful tool for studying heterogeneous populations in various fields, including biology, medicine, economics, and engineering. However, fitting these models to data is computationally challenging if the description of individuals is complex and the population is large. To address this issue, we propose a novel machine learning-based approach: We exploit neural density estimation based on normalizing flows to approximate individual-specific posterior distributions in an amortized fashion, thereby allowing for an efficient inference of population parameters. Applying this approach to problems from cell biology and pharmacology, we demonstrate its scalability to large data sets in an unprecedented manner. Moreover, we show that it enables accurate uncertainty quantification and extends to stochastic models, which established methods, such as SAEM and FOCEI are unable to handle. Thus, our approach outperforms state-of-the-art methods and improves the analysis capabilities for heterogeneous populations.

1 Introduction

Heterogeneity within populations is a common phenomenon in various fields, including epidemiology, pharmacology, ecology, and economics. It is, for instance, well established that the human immune system exhibits substantial variability among individuals [1, 2], that individual patients respond differently to treatments [3–5], that genetically identical cells develop pronounced cell-to-cell variability [6, 7], but also that individual students perform differently depending on their socioeconomic group and school [8]. This heterogeneity can be described and analyzed using *non-linear mixed-effects (NLME)* models, a powerful class of statistical tools. NLME models can account for similarities and differences between individuals using fixed effects, random effects, and covariates. This allows for a high degree of flexibility and interpretability. These models are widely used for statistical analysis [9, 10], hypothesis testing [11], and predictions [3, 4].

NLME models depend on unknown parameters, such as reaction rates and initial concentrations, which need to be estimated from data. Indeed, parameter estimation – often also called *inference*

*jan.hasenauer@uni-bonn.de

– provides key insights about the data and underlying processes. The main challenge in inferring these parameters lies in the required marginalization over random effects at the individual level. For this, there is generally no closed-form solution [12]. Particularly for large populations, this becomes a problem, as the marginalization must be performed for all individuals.

The most frequently used inference methods at present are deterministic, starting from the first inference method introduced by Beal & Sheiner based on a first-order approximation of the model function around the expected value of random effects [13] and later on conditional modes [14]. It was used, among others, to analyze clinical patient data [15]. Pinheiro & Bates reviewed more accurate methods based on the approximation of the marginal likelihood using Laplace methods or quadrature rules, which can provide higher accuracy, but also come with higher computational costs [16]. Today, first-order conditional estimation with interaction (FOCEI) [17] is arguably the most common inference method used in pharmacokinetic modeling. However, the aforementioned methods have statistical drawbacks, as they do not necessarily converge to maximum likelihood estimates, and estimates can be substantially biased when the variability of random effects is large [18, 19]. For unbiased results, Kuhn & Lavielle [20] introduced a stochastic expectation maximization algorithm (SAEM), which converges under very general conditions [20]. This method was applied, for example, to model the response of yeast cells to repeated hyperosmotic shocks [10]. Yet, the algorithm can be computationally demanding, especially for models with a large number of random effects and models with complex structures. In addition to the aforementioned frequentist approaches, Bayesian methods applied at the population level have been proposed (see the review [21]), which are even more computationally demanding, but inherently facilitate uncertainty quantification. To accelerate inference with sampling algorithms for NLME models Augustin *et al.* used a simulation-based approach [22]. However, all the methods mentioned do not apply to stochastic models for the individual, such as stochastic differential equations (SDEs). So far only Bayesian methods can provide exact inference for SDEs, with high computational costs [23, 24]. In general, computational costs make it difficult to fit NLME models to large data sets, for example, thousands of cells in single-cell experiments or large cohorts of patients, and to obtain reliable estimates of model parameters [22, 25, 26]. Furthermore, multiple starts of the estimation procedure are needed, further increasing computational costs, as parameter estimation can be sensitive to the choice of initial parameter values, making it difficult to find global maximum likelihood estimates [12].

Here, we present an alternative approach based on invertible neural networks to estimate the parameters of NLME models. We use simulation-based neural posterior estimation, which has been developed to address general parameter estimation problems [27, 28]. We train a mapping – a conditional normalizing flow parameterized by an invertible neural network – from a latent distribution to individual-specific posteriors conditioned on observed individual-level data. During training of this neural posterior estimator, only simulations from a generative model are used. In the latter inference phase, the trained estimator can be applied highly efficiently to any data set with similar measurements and different population models without any further model simulations, facilitating the estimation of NLME model parameters in an amortized fashion. We compare our method with state-of-the-art and widely used methods in the field of NLME models on problems from cell biology and pharmacology: the stochastic approximation expectation maximization algorithm (SAEM) [20] implemented in `Monolix` [29] and the first-order conditional estimation with interaction (FOCEI) [17] implemented in `NONMEM` [13].

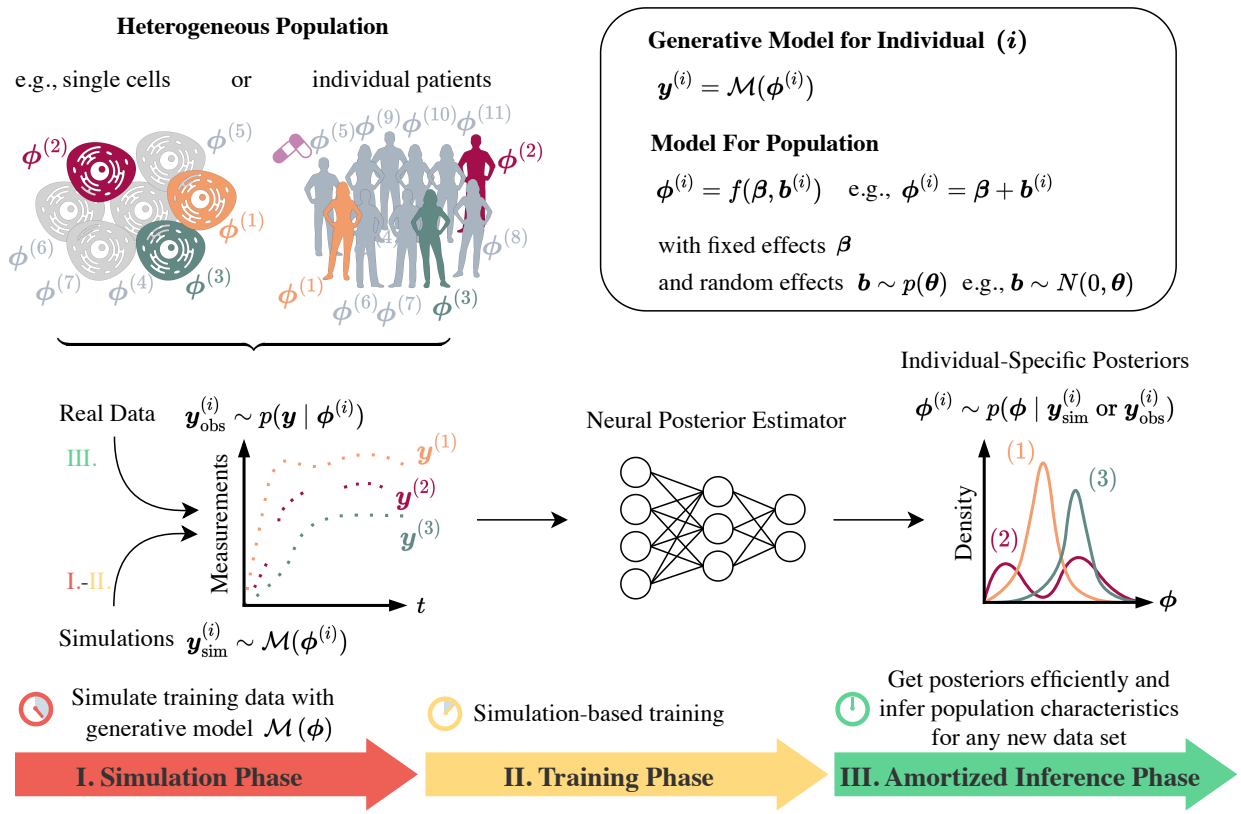


Figure 1: *Three phases of the amortized approach.* (**I.**) The simulation phase, where we generate data from the model $\mathcal{M}(\phi)$, (**II.**) the training phase, where we train the neural posterior estimator to predict individual-specific posteriors based on the simulations, and (**III.**) the amortized inference phase, where we infer the population parameters of the non-linear mixed-effects model given observed data.

2 Results

2.1 An amortized machine learning-based approach to fit NLME models

To facilitate scalable and flexible parameter estimation for NLME models, we developed and implemented an approach based on amortized machine learning. The approach allows inferring the parameters of NLME models with deterministic and stochastic mathematical models for individuals. In practice, individuals are often modeled using ordinary (ODE) or stochastic (SDE) differential equations. Such models typically depend on unknown parameters $\boldsymbol{\phi} \in \mathbb{R}^k$, such as reaction rates or initial concentrations, that need to be inferred from data. We assume that the underlying data generation process can be described via a mechanistic model $\mathcal{M}(\boldsymbol{\phi})$, incorporating dynamics, intrinsic sources of stochasticity, as well as measurement noise. We consider a set of measurements $\mathcal{D} = \{\boldsymbol{y}^{(i)}\}_{i=1}^{n}$ with $\boldsymbol{y}^{(i)} \in \mathbb{R}^{n_i}$ for individuals i, e.g., measurements for different cells or patients. To account for population heterogeneity, we assume that each individual i can be described by parameters $\boldsymbol{\phi}^{(i)}$, which consist of *fixed effects* $\boldsymbol{\beta}$ shared across the population, and/or *random effects* $\boldsymbol{b}^{(i)}$ specific to individuals. This relation is described by a *population model* $\boldsymbol{\phi}^{(i)} = f(\boldsymbol{\beta}, \boldsymbol{b}^{(i)})$. As usual in practice, we fully characterize the distribution of individual-specific parameters $\boldsymbol{\phi}^{(i)}$ via population parameters $\boldsymbol{\theta}$, e.g., $\boldsymbol{\phi}^{(i)} = \boldsymbol{\beta} + \boldsymbol{b}^{(i)}$ and $\boldsymbol{b}^{(i)} \sim \mathcal{N}(0, \boldsymbol{D})$ with covariance matrix $\boldsymbol{D}$, we

write as $\boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{D})$. Together, this defines a *non-linear mixed-effects* (NLME) model.

In order to estimate the population parameters $\boldsymbol{\theta}$, the joint likelihood of the data $\mathcal{D}$ given $\boldsymbol{\theta}$,

$$p(\mathcal{D} \mid \boldsymbol{\theta}) = \prod_{i=1}^{n} \int p(\boldsymbol{y}^{(i)} \mid \boldsymbol{\phi}) p(\boldsymbol{\phi} \mid \boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\phi}, \tag{1}$$

is maximized. The likelihood $p(\boldsymbol{y}^{(i)} \mid \boldsymbol{\phi})$ is implicitly induced via the generative model $\mathcal{M}$ and the conditional density $p(\boldsymbol{\phi} \mid \boldsymbol{\theta})$ is defined by the chosen population model. The maximization is computationally demanding, as it involves marginalization over unobserved random effects. Usually, the integral has no closed-form solution and even the likelihood $p(\boldsymbol{y}^{(i)} \mid \boldsymbol{\phi})$ may be intractable, as is, for example, the case for stochastic models. Established methods need a tractable likelihood and approximate the integral for each individual, either by linearization around the modes of the integrand conditioned on the population parameters (such as FOCEI) [17], or by sampling individual-specific parameters conditioned on the observations of the individuals and the population parameters (such as SAEM) [20]. Both approaches work in an iterative manner, where alternately the individual-specific parameters and the population parameters are optimized.

We note that the marginal likelihood (1) can be written as a conditional expectation over individual-specific posteriors $p(\boldsymbol{\phi} \mid \boldsymbol{y})$ given a prior $p(\boldsymbol{\phi})$ that is non-zero on the integration domain,

$$p(\mathcal{D} \mid \boldsymbol{\theta}) = \prod_{i=1}^{n} p(\boldsymbol{y}^{(i)}) \int p(\boldsymbol{\phi} \mid \boldsymbol{y}^{(i)}) \frac{p(\boldsymbol{\phi} \mid \boldsymbol{\theta})}{p(\boldsymbol{\phi})} \, \mathrm{d}\boldsymbol{\phi} = \prod_{i=1}^{n} p(\boldsymbol{y}^{(i)}) \mathbb{E}_{\boldsymbol{\phi} \sim p(\boldsymbol{\phi} \mid \boldsymbol{y}^{(i)})} \left[\frac{p(\boldsymbol{\phi} \mid \boldsymbol{\theta})}{p(\boldsymbol{\phi})} \right]. \tag{2}$$

This means that samples from individual-specific posteriors would facilitate the construction of a Monte Carlo estimator for the population-level marginal likelihood. Thus, we obtain optimal population parameters $\boldsymbol{\theta}^*$ by taking the logarithm of (2), which is commonly done for numerical stability [30], and solving the minimization problem

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} - \log p(\mathcal{D} \mid \boldsymbol{\theta}) \approx \arg\min_{\boldsymbol{\theta}} - \sum_{i=1}^{n} \log \left(\frac{1}{M} \sum_{j=1}^{M} \frac{p(\boldsymbol{\phi}_j^{(i)} \mid \boldsymbol{\theta})}{p(\boldsymbol{\phi}_j^{(i)})} \right), \tag{3}$$

with $\boldsymbol{\phi}_j^{(i)} \sim p(\boldsymbol{\phi} \mid \boldsymbol{y}^{(i)})$ i.i.d. for $j = 1, \ldots, M$ for each individual i.

Based on these insights, we present here a novel three-phase procedure for the inference of NLME models (Figure 1): (I) In the simulation phase, we use the generative model $\mathcal{M}(\boldsymbol{\phi})$ and multiple samples $\boldsymbol{\phi}$ from the prior $p(\boldsymbol{\phi})$ to produce a set of simulations $\{\boldsymbol{y} \sim \mathcal{M}(\boldsymbol{\phi})\}$. (II) In the training phase, we learn a global approximation $q(\boldsymbol{\phi} \mid \boldsymbol{y}) \approx p(\boldsymbol{\phi} \mid \boldsymbol{y})$ for any $(\boldsymbol{\phi}, \boldsymbol{y}) \sim p(\boldsymbol{y} \mid \boldsymbol{\phi}) \cdot p(\boldsymbol{\phi})$, with q parameterized as a normalizing flow via an invertible neural network [27]. The approximation q is trained using the generated pairs of parameters and synthetic data $(\boldsymbol{\phi}, \boldsymbol{y})$ to minimize the Kullback-Leibler divergence between the true and approximate posterior distributions for any $\boldsymbol{y}$. Instead of inserting data $\boldsymbol{y}$ directly into the invertible neural network as a conditional input, we use summary networks, such as vision or sequence models, to reduce the dimension of the data [27]. The summary and invertible network can be trained jointly, and we check the approximation quality by calibration diagnostics. (III) After sufficiently long simulation and training phases with simulated data, we obtain a global approximation of the true posterior distribution from which we can efficiently draw samples conditioned on so far unseen data. In the amortized inference phase, we assume a population model and infer the population-level parameters $\boldsymbol{\theta}$ using the approximation to the population likelihood (3) based on samples from individual-specific posterior distributions. This likelihood is amenable and minimized using a gradient-based optimizer. The minimization is computationally efficient and simple, since only sampling from the posterior distributions is required.

In summary, we split the inference of population parameters into data-free simulation and training phases during which we learn a global posterior approximation, and an efficient inference phase during which we no longer need to simulate the potentially expensive mechanistic model, but simply sample from the trained neural posterior estimator. Owing to its low computational cost, the inference phase can be, e.g., easily repeated for different population models to perform model selection, and can handle multiple data sets with many individuals. Thus, in these cases, we amortize the cost of training the neural posterior estimator.

2.2 Normalizing flows provide accurate and efficient approximation of individual-specific posteriors

The proposed approach to fitting the NLME models is based on the approximation of the individual-specific posterior distributions with normalizing flows, which are learned in the training phase. As the accuracy of these approximations is critical, we assessed in a first step the approximation quality. Therefore, we considered two published ODE-based NLME models of mRNA transfection [25]. These ODE models describe the transfection process (Figure 2A) – which is at the core of modern mRNA vaccines [31] – at the single-cell level. The models possess, respectively, 6 and 11 parameters that describe 2 and 4 hidden state variables (Figure 2B, see Supplement A.1 for details on the models). Single cells were transfected with mRNA coding for a green fluorescent protein (GFP), and dense temporally resolved fluorescence intensities of different cells were measured for 30 hours using micropatterned protein arrays and time-lapse microscopy (Figure 2C).

We verified the accuracy of our neural posterior estimator using simulation-based calibration (SBC) plots [32], and compared the posterior estimates obtained using our method with reference methods for randomly chosen synthetic and real single cells, in particular using Markov chain Monte Carlo (MCMC) with adaptive parallel tempering implemented in `pyPESTO` [33]. We found that for both ODE models, the SBC plots show no systematic bias, and the neural posterior estimator matches the MCMC posteriors well, indicating that the individual-specific posteriors in different parameter regimes for both synthetic and real samples from the data were accurately captured (Supplement Figures A3, A4, A5). Furthermore, the posterior fit at the single-cell level demonstrates a high level of accuracy (Figure 2D).

An assessment of computation time revealed that the employed MCMC sampler required approximately 1 million samples and 10 chains with an effective sample size of 195, which took around 20 minutes of computation time for a single cell. In comparison, the trained neural posterior estimator only required a few seconds for the same effective sample size and on the same set-up (see details on the implementation in Methods 4.5). Thus, in this case, the training time of the neural networks to obtain individual-specific posteriors, ~ 6.5 hours, would be amortized after around 20 cells, or even after an individual cell if a sufficiently high sample size is required. This demonstrates the efficiency of neural posterior estimation for parameter estimation also outside a mixed-effects context.

2.3 Machine learning-based approach provides accurate estimates of population parameters

Given the accurate approximation of posteriors on an individual-specific level, we can use the pre-trained densities to estimate the NLME population parameters. To assess the accuracy of our approach, we generated synthetic data using the two NLME models of mRNA transfection (see Supplement A.1.2), and compared the mean squared distance of the true parameters to the estimated parameters of our approach to the estimated parameters of the state-of-the-art method SAEM [20]

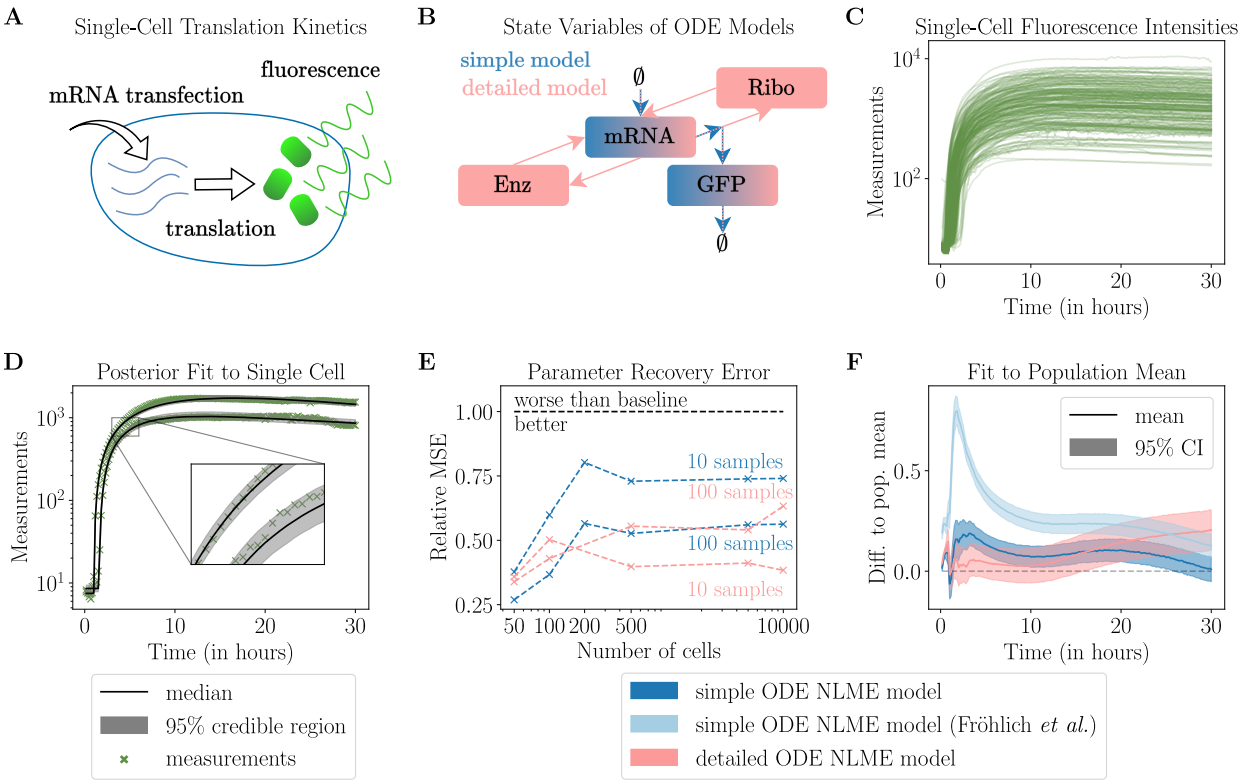


Figure 2: *Validation of the amortized approach on single-cell NLME models.* (**A**) Single-cell translation kinetics after mRNA transfection. (**B**) Visualization of the simple and detailed single-cell ODE models, where the color refers to the states included in the respective model (see Supplement A.1 for details on the models). (**C**) Fluorescent intensity time courses of 200 single cells (first out of 5488). (**D**) Credible regions of trajectories (simple single-cell ODE model) estimated by the neural posterior estimator for two real cells. (**E**) Median of the mean squared error (MSE) of the estimated compared to the true parameters of the synthetic data for both single-cell NLME models is shown for different numbers of cells and numbers of posterior samples $M = 10, 100$ used in the Monte Carlo approximation (median of the best 10 multi-starts divided by the minimal error achieved by the baseline method). (**F**) The difference in the population mean estimated from real trajectories and simulations generated with the estimated population parameters is shown with a 95% confidence interval (CI). Additionally to the single-cell models fitted with the amortized approach, the best fit of Fröhlich *et al.* for the simple ODE model is shown [25].

implemented in `Monolix` [29], which is unbiased and converges under very general conditions [20]. As the SAEM estimates depend on the starting point, we performed a multi-start using 100 different starting points (sampled from the same prior as used in the training phase of the neural posterior estimator). Moreover, we compared our results with those published in [25], where a Laplacian approximation together with a multi-experiment setup on real data was introduced to improve parameter identifiability (see Supplement A.1.1).

Our experiments show that, for different data set sizes and models, our method was able to recover the true parameters with a lower recovery error than SAEM (Figure 2E). For each ODE model, we trained only one neural posterior estimator, which could be used for inference on all different single-cell data sets, while SAEM needed a full restart for each data set. In addition, the estimated population mean of the simple model of the machine learning-based approach shows

a better fit of the population mean compared to the results published in [25] for the real data (Figure 2F). Furthermore, we can confirm the result of [25] that the detailed model describes the initial fluorescence activity more accurately (Figure 2F).

In summary, our approach based on amortized neural posterior estimation was able to provide accurate estimates of population parameters for synthetic and real data. Moreover, we needed only one neural posterior estimator to be trained for each model and we could apply it to synthetic and real data sets of different sizes.

2.4 Amortization for large populations, new data sets and changing population models achieved

As the computational cost of the state-of-the-art method SAEM increases linearly with the number of individuals in a population (Figure 3A), we compared the computation time for the estimation of population parameters of our machine learning-based approach to SAEM [20] implemented in `Monolix` [29].

The assessment of the overall computation times revealed that the computationally demanding phase in our approach is the data simulation using the mechanistic model and the training of the neural posterior estimator. For both phases, the detailed NLME mode required three times as much computation time compared to the simple model. Afterwards, inferring the population parameters for a particular new data set can be done highly efficiently within seconds. Our method scales nearly constantly with respect to the number of individuals in the population (Figure 3A). In particular, if the population was large ($10,000$ cells in the case of the single-cell NLME models), we already amortized the training time cost compared to SAEM for a single data set.

Before, the parameters in the single cell NLME models were assumed to be independently distributed. However, cross-correlations between parameters are essential to explain population behavior [10], but were not captured in [25] due to computational costs. Indeed, for the detailed mRNA transfection model, the medians of the individual-specific posteriors of the respective parameters show a clear correlation (Figure 3B). So, instead of assuming a diagonal covariance matrix for the random effects, we changed only the population model to allow for a full covariance matrix and repeated the amortized inference phase without any further training of the neural posterior estimator. Including these correlations substantially improved the fit of the population variance (Figure 3C), which confirms the findings on the importance of incorporating cross-correlations between parameters in [10]. Moreover, while applying our approach to real data in the multi-experiment setup of [25], we already effectively changed the population model compared to the synthetic data setting, as most parameters were assumed to be shared between multiple experiments. To account for this, we included an indicator function to map experiment-specific and population parameters onto cell-specific parameters (see Supplement A.1.1).

In summary, our analyses showed that our approach scales to large populations and allows for the reuse of the trained neural posterior estimator on different data sets and for different population models at almost no additional computational cost, rendering it substantially more scalable than state-of-the-art methods.

2.5 Robust uncertainty analysis becomes possible due to efficient inference of the population model

Our previous evaluations have shown that the approach based on amortizing neural posterior estimation allows efficient construction of point estimates. Beyond point estimates, in many applications, it is important to assess the uncertainty of the parameters, e.g., to determine the

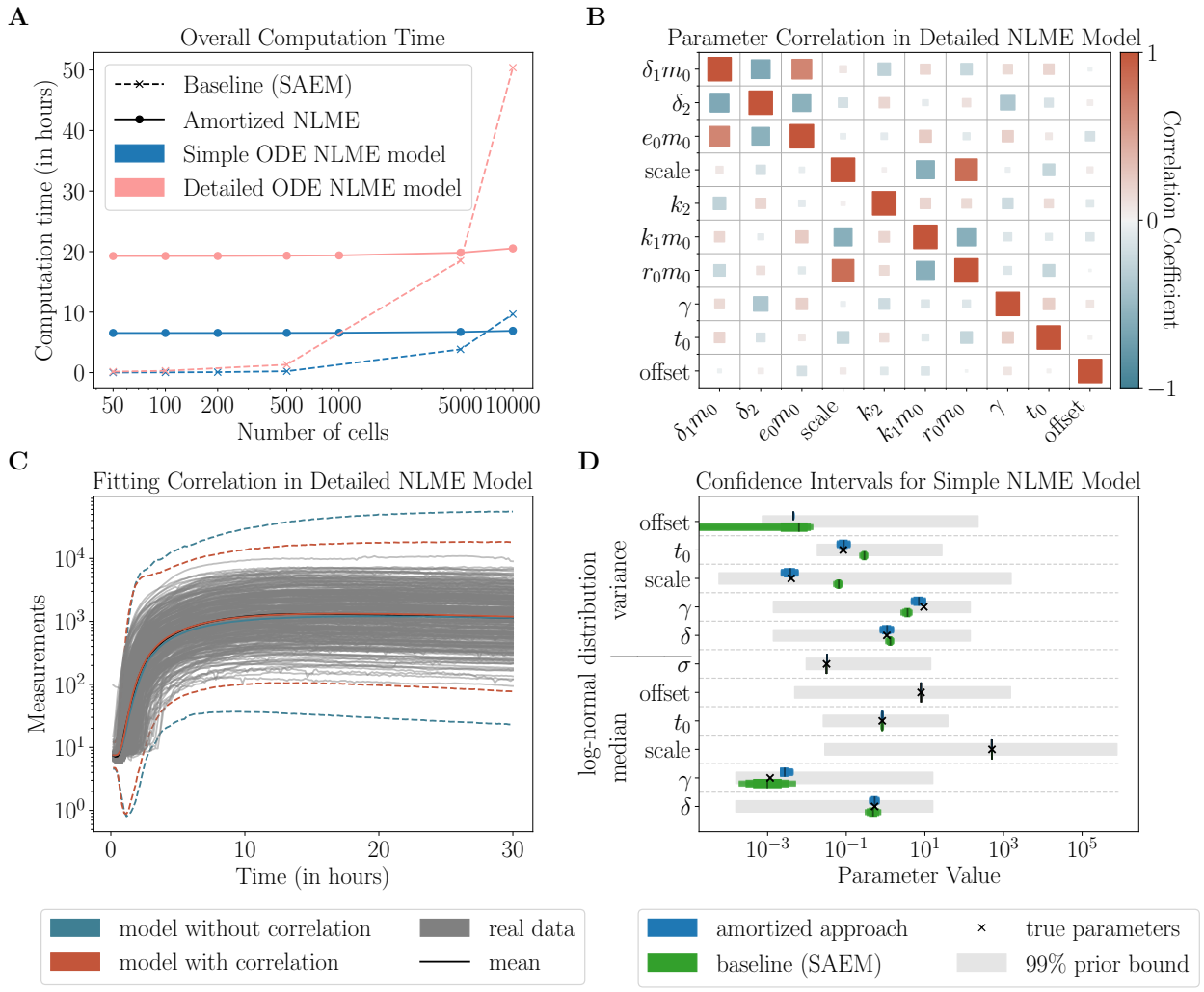


Figure 3: *Flexibility and scalability of the amortized approach on the single-cell NLME models.* (**A**) Overall computation time (average over ten best multi-starts) for the single-cell NLME models compared to the baseline using parallelization. (**B**) Inter-individual correlation of the parameters in the detailed single-cell model. Size and color of the boxes represent the estimated correlation between the medians of the posterior distributions given by the neural density estimator for the respective parameters on the real data. (**C**) Mean and 99% confidence intervals of the simulations for the detailed NLME model, where the population parameters are assumed to be log-normally distributed with and without correlations between parameters. (**D**) 80%, 90%, and 95% confidence intervals (CIs) for the simple single-cell NLME model (see Supplement Figure A1 for the other models) using synthetic data with known true parameters. The true variance of the offset is 0 and, therefore, cannot be seen.

identifiability of the parameters, draw reliable conclusions, and make representative predictions [30, 34]. The implementation of SAEM in `Monolix` allows standard errors to be obtained through linearization of the likelihood or by a stochastic approximation of the Fisher information matrix, which yields asymptotically correct results under the assumption of normally distributed errors and a large amount of data. Using these standard errors, the confidence intervals are calculated using the Wald statistic [29]. However, to ensure the validity of the confidence intervals, it is often advisable

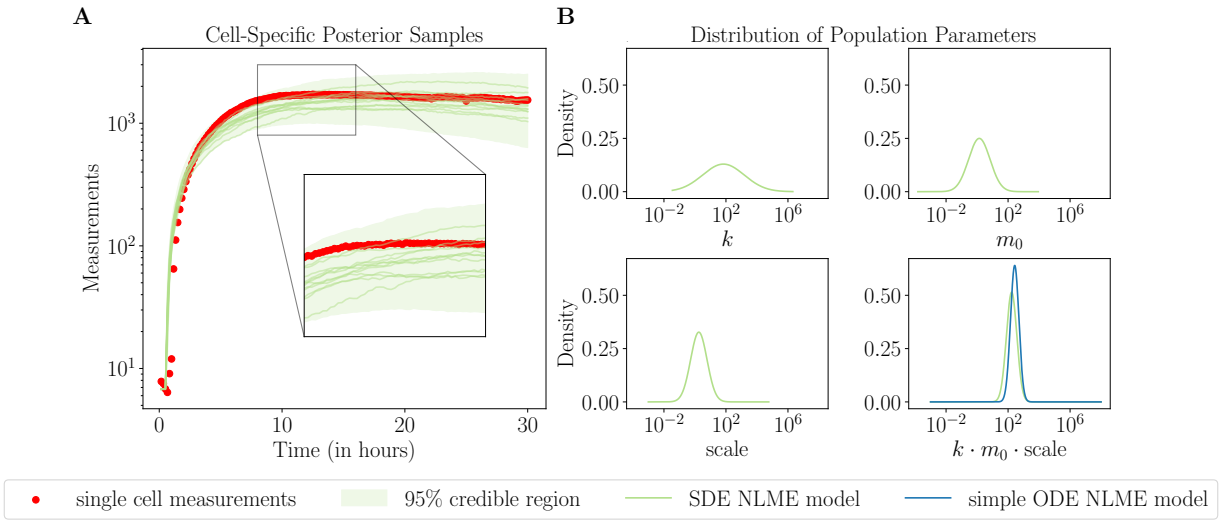


Figure 4: *Stochastic NLME model improves identifiability compared to deterministic counterpart.* (**A**) Credible regions of a trajectory of the SDE single-cell model estimated by the neural posterior estimator for a real cell. The estimated median of the posterior was simulated 10 times. (**B**) Estimated population distributions for the parameters k, m_0 and scale for the SDE NLME model and their product in the simple ODE NLME model.

to use bootstrapping or non-local approaches such as profile likelihoods, as these are more accurate when the above assumptions are not met, e.g., allowing non-symmetric confidence intervals [35]. These methods are not supported directly in `Monolix`. Moreover, such tests are infeasible when the computational time is too high, as is the case with SAEM, or biased, when the estimates are already biased, as can be the case with FOCEI.

Given the computational efficiency of the inference phase in our approach, we explored the possibility of performing accurate uncertainty quantification. Specifically, as it is a widely used non-local frequentist approach to uncertainty quantification in system biology, we applied profile likelihood analysis [36]. This revealed that the computation of profile likelihoods takes only seconds, whereas linearization with SAEM already takes on the order of minutes. In this case, the confidence intervals based on the profile likelihoods were comparable to those based on linearization using SAEM for most parameters. Yet, for three variance parameters, the 80% CIs computed with SAEM actually do not cover the true parameter, while the CIs computed with profiles from the amortized approach do (Figure 3D).

In conclusion, our amortized approach allows for an effective and robust uncertainty quantification. In principle, in addition to computing profile likelihoods, bootstrapping could also be easily done using the amortized approach since the training and simulation phase is data-free. This is a key advantage, as other frequentist methods do not allow for a robust uncertainty analysis due to substantially higher computational costs.

2.6 Stochastic mixed-effects models become easily tractable

As our approach based on neural posterior estimation proved to be valuable for deterministic models, we assessed its capability to cope with stochastic models, which often provide a more adequate description of the underlying process [37, 38]. At the single cell level, ignoring the inherent stochastic nature of reactions can bias parameter estimates [23], and pooling measurements from several cells is

indispensable for reliable estimates [39]. However, for such models, the likelihood function – which our purely simulation-based approach does not need – is often unavailable, requiring computationally demanding techniques such as approximate Bayesian computation or a Metropolis-within-Gibbs algorithm, which can handle the unavailable likelihood function via correlated particle filters [23, 24, 40]. Here, we again considered the processes of mRNA transfection, but described by a stochastic differential equation (SDE) as proposed by [41] (see the model specification in Supplement A.1). This model has been shown to be superior for the description of individual cells and to improve parameter identifiability [41], but has not been used so far in an NLME modeling framework.

The evaluation using the SDE NLME model on synthetic data revealed that the machine learning-based approach was indeed able to accurately recover the stochastic NLME model parameters (Supplement Figure A1). Moreover, the posterior fit for a single real cell is accurate (Figure 4A). Further analysis on synthetic data generated by the SDE NLME model showed that the simple ODE NLME model estimated parameters such that the variance of the population was 3 times larger than the true variance, while for the stochastic NLME model the variance is only 1.3 times larger and hence capable of capturing the data more accurately (Supplement Figure A2). This, in particular, underlines that a deterministic model can give erroneous results if it inadequately captures the underlying processes. The overall computational time (18 hours) was comparable to the detailed ODE model used before (19 hours), and the amortized inference phase remained highly efficient.

The simple ODE model of the mRNA transfection processes possessed structural non-identifiabilities, meaning that not all the parameters can be determined from the data. Consequently, the ODE model encompasses only the product $k \cdot m_0 \cdot \text{scale}$, while the SDE model encompasses the individual parameters k, m_0 and scale, offering a more detailed representation. Indeed, using our amortizing NLME framework, we were able to identify all parameters of the stochastic NLME model (Figure 4B).

In summary, stochastic models can be a more accurate description of the underlying process, and our machine learning-based approach enables the use of either a deterministic or a stochastic NLME model, whichever is more appropriate. This enables not only a more profound understanding of the actual mechanism, but can also improve model identifiability.

2.7 Individual-specific characteristics can be handled

So far, we have considered inference problems in which all individuals (or at least batches thereof) were subject to similar conditions. However, in practice, there are often further individual-level characteristics – *covariates* – available, such as age, dosing regimes, or preconditions of patients. In pharmacokinetics, one is interested in describing the absorption and distribution of drugs within the body, and usually some characteristics of individuals are known, but measurements are often sparse (Figure 5A). These characteristics pose a challenge for simulation-based algorithms that require training data similar to the data of interest. Therefore, we studied the applicability of our approach to a pharmacokinetic ODE model as introduced in [42], describing the distribution of an angiogenesis inhibitor, a drug that inhibits the growth of new blood vessels, and its metabolite in a compartmental model. We used measurement data (sunitinib and SU12662 plasma) from a cohort of 47 patients, including covariates such as age, sex, and medication times and quantities (see Supplement A.2). As it is arguably the most common inference method used in pharmacokinetic modeling, here we considered FOCEI [17] for comparison, implemented in `NONMEM` [13]. We compared the simulations for each individual generated from the estimated parameters of both methods. To simulate individual patients from so-called empirical Bayes estimates, we fixed the population parameters and maximized for each patient individually the scaled posterior $p(\boldsymbol{\phi} \mid \boldsymbol{y}^{(i)})/p(\boldsymbol{\phi})p(\boldsymbol{\phi} \mid \boldsymbol{\theta})$ or, as usual, the scaled likelihood $p(\boldsymbol{y}^{(i)} \mid \boldsymbol{\phi})p(\boldsymbol{\phi} \mid \boldsymbol{\theta})$ using FOCEI's linearization of the likelihood.

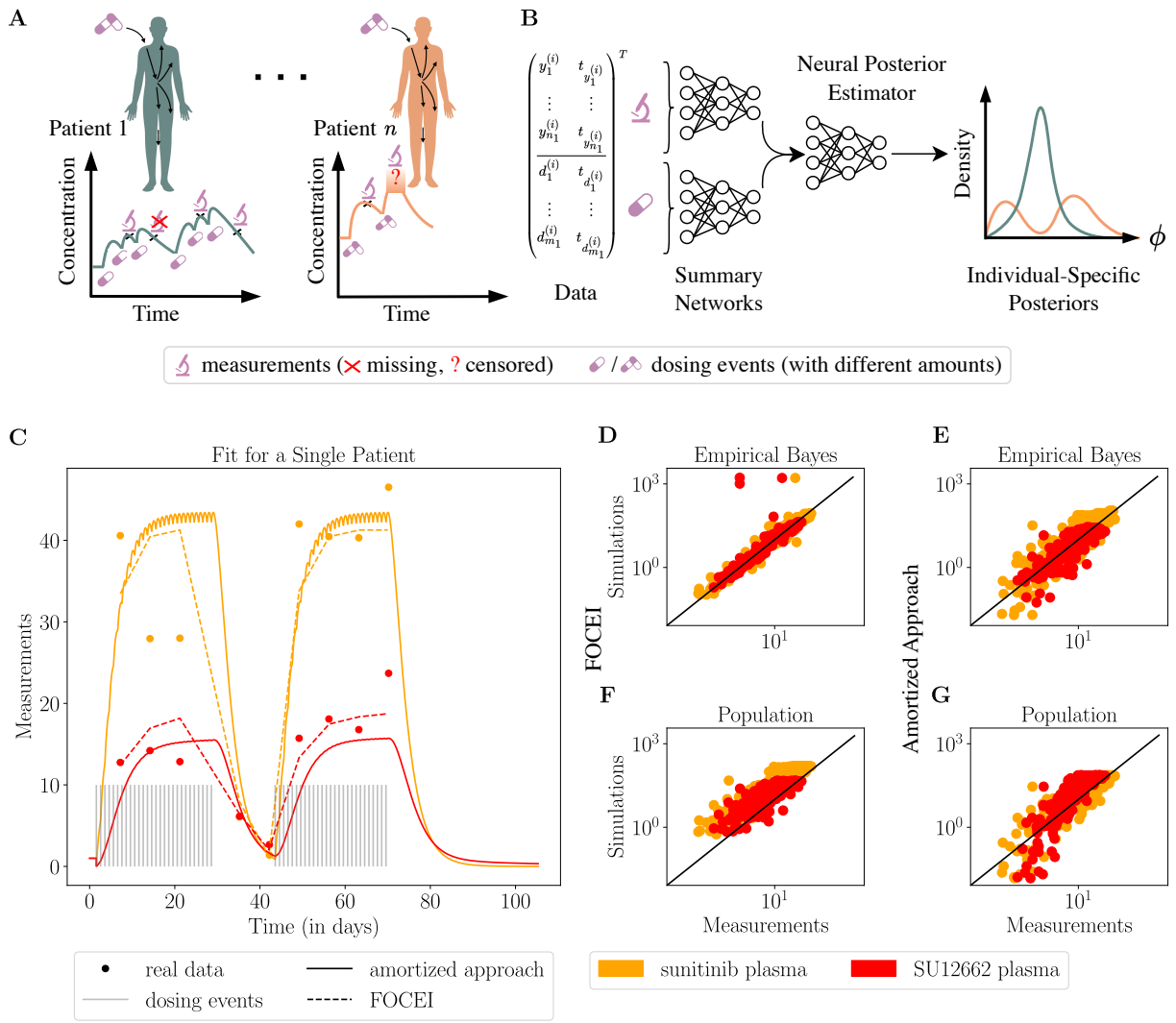


Figure 5: *Amortization in pharmacokinetic modeling.* (**A**) Visualization of patient data: individual-specific measurements and dosing regimes. (**B**) Encoding of dosing events as part of observations, which are given to the summary network of the neural posterior estimator to estimate individual-specific posterior distributions. (**C**) Fit of a single patient using FOCEI and the amortized approach to NLME models. (**D**–**E**) Measurements against simulations of empirical Bayes estimates using FOCEI (**D**) and the amortized approach (**E**), respectively. (**F**–**G**) Measurements against simulations of estimated population parameters (excluding random effects) using FOCEI (**F**) and the amortized approach (**G**), respectively.

In our amortizing framework, covariates such as age can be treated either as random variables on the individual level or as part of the population model. If they are part of the population model, the covariates can be mapped to the random effects and can partially explain them. If they are instead part of the model $\mathcal{M}$, then they need to be synthetically generated during the simulation phase. This is the case with dosing regimes, which refer to the prescribed schedules and dosages of medications that are administered to patients. Therefore, we encoded the dosing events as part of the observations, which are given to the summary network (Figure 5B). During the simulation, we

generated the dosing events stochastically from a reasonable prior range.

Analyses of the amortized approach to fit this pharmacokinetic NLME model revealed that simulating data now took considerably more time (56 hours on 8 cores) due to model complexity and discrete events. On this small data set, FOCEI was much faster and needed only a few minutes. After training, inferring the population parameters using our approach took only seconds due to the small cohort of patients.

Our approach was able, similar to FOCEI, to fit the measurements for a single patient (Figure 5C). Comparing the results of our machine learning-based approach with those obtained using FOCEI revealed that the simulations for individual patients, the empirical Bayes estimates, generally seem to match the measurements of that individual better than the machine learning-based approach (Figure 5D–E). Yet, correlation of simulations and measurements is higher for the amortized approach (0.752 for sunitinib and 0.746 for SU12662) than for FOCEI (0.235 and 0.006, respectively), since FOCEI has several severe outliers, which our method does not have (Figure 5D–E). Simulations at the population level (i.e., random effects were set to 0) show that the ones generated by FOCEI tend to be larger than the actual measurements, while the difference between the measurements and simulations from our approach based on neural posterior estimation is more symmetrically distributed (Figure 5F–G). Thus, FOCEI appears to give a biased estimate of the population, which is a known problem for this deterministic method [18, 19, 43]. However, the correlation of simulations of the population and measurements is similar for both approaches: 0.806 (sunitinib) and 0.743 (SU12662) for FOCEI and 0.779 and 0.748 for the amortized approach, respectively.

This proof-of-concept application demonstrates that our method is able to handle individual-level covariates such as dosing regimes. In particular, we observed a less biased population fit as compared to FOCEI. However, further research is needed to, e.g., calibrate the summary network and hence improve the fit of the individual patients. For larger cohorts of patients, we would expect to see efficiency advantages compared to FOCEI also in the overall computing time, which however remains to be investigated.

3 Discussion

We developed a novel approach to non-linear mixed-effects model inference based on amortized neural posterior estimation. The proposed method offers several advantages such as scalability, flexibility, and accurate uncertainty quantification over established approaches, as we demonstrated on problems from single-cell biology and pharmacology.

One of the most important benefits of the method is its scalability. The efficient amortizing inference phase allows to scale to large numbers of individuals and can be applied to before unseen data. The whole workflow scales almost constantly in the number of individuals in the data. The main bottleneck, the simulation and training phases, can be tackled by more extensive parallelization on a high-performance infrastructure, since all simulations are independent. Further, the method can be applied to various population models with low computational costs using the same trained neural posterior estimator, allowing efficient model selection. In contrast, state-of-the-art methods require a full restart for each population model. In addition, our approach allows to flexibly incorporate individual-specific characteristics enabling an efficient selection of covariates in the population model. Our machine learning-based approach is purely simulation-based; that is, it does not require the evaluation of likelihoods, but only a generative model to simulate synthetic data. Therefore, it can be easily used even for complex stochastic models, which established approaches fall short of, as we demonstrated on an SDE-based NLME model of mRNA transfection. This can be easily extended to Markov jump processes, e.g., simulated with the Gillespie algorithm [44]. This generality is unique in

the NLME context, as special frameworks needed to be developed to cope with stochastic differential equations [23] or Markov jump processes [39]. Lastly, the efficient neural posterior estimator facilitates the use of more accurate and systematic methods to assess parameter uncertainty. Here, we demonstrated this by combining our approach with profile likelihoods as a proof-of-concept, but other approaches, such as bootstrapping and Bayesian sampling, could conceptually also be efficiently applied.

Despite its benefits, the proposed method has some limitations. For small data sets and if no population model selection or accurate uncertainty quantification is performed, the computation time of our approach, its simulation and training phases, will be higher compared to established methods. Additionally, the proposed method may produce erroneous parameter estimates if the prior is too narrow or if the underlying model is misspecified [45], or it may produce non-conservative posterior estimates [46]. However, misspecification of the model is a general problem for state-of-the-art methods as well. A solution might be to extend the loss function during training to include a misspecification measure [45]. On the other hand, the accuracy of the approximated posteriors can be checked after training, e.g., by simulation-based calibration [32], or individual posterior checks by MCMC or approximate Bayesian computation (ABC) [40]. These, however, introduce an additional computationally expensive step. Imperfect approximations of true posteriors can occur if the conditional normalizing flows, our foundation of the global posterior estimator, are not expressive enough [47]. This might be the case for multimodal distributions in general [47], but not for the examples we considered. Nevertheless, the approximations could be improved by a deeper architecture, or one could consider generalized normalizing flows [47], conditional variational autoencoders [48] or conditional generative adversarial neural networks [49] as basis of the global posterior estimator.

In conclusion, the amortized approach we presented in this study offers a powerful solution for non-linear mixed-effects modeling, enabling researchers to flexibly use models for individuals – including stochastic ones – and the population while performing accurate parameter estimation and uncertainty analysis, and to gain a deeper understanding of the underlying processes in a more scalable manner than state-of-the-art methods.

4 Methods

Non-linear mixed-effects models are a powerful statistical tool for analyzing data that are both clustered and non-linear. Here we will present the three phases of the amortized approach to NLME models starting from the individual level going to the population.

4.1 The generative model

We consider a set of observed i.i.d. data $\mathcal{D} = \{\boldsymbol{y}^{(i)}\}_{i=1}^{n}$ from a population. These measurements per individual can be made at different times, with different recurrences (including snapshot measurements) and of different dimensions n_i. We assume that an individual $\boldsymbol{y}^{(i)} \in \mathbb{R}^{n_i}$ can be described through a generative process $\mathcal{M}(\boldsymbol{\phi})$ with unknown parameters $\boldsymbol{\phi} \in \mathbb{R}^k$. As a generative model, we understand any parametric model, such as linear models, differential equations, or Markov jump processes, which can produce predictions of our observables $\boldsymbol{y}^{(i)}$ for an individual i given some parameters $\boldsymbol{\phi}$ (see Supplements A.1 and A.2 for the models used in this manuscript). Since measurements are noisy, noise generation is part of the generative model; e.g., normal noise is added after simulation of the data. The first phase of our method is to simulate training data from this model using samples from a parameter prior distribution $p(\boldsymbol{\phi})$.

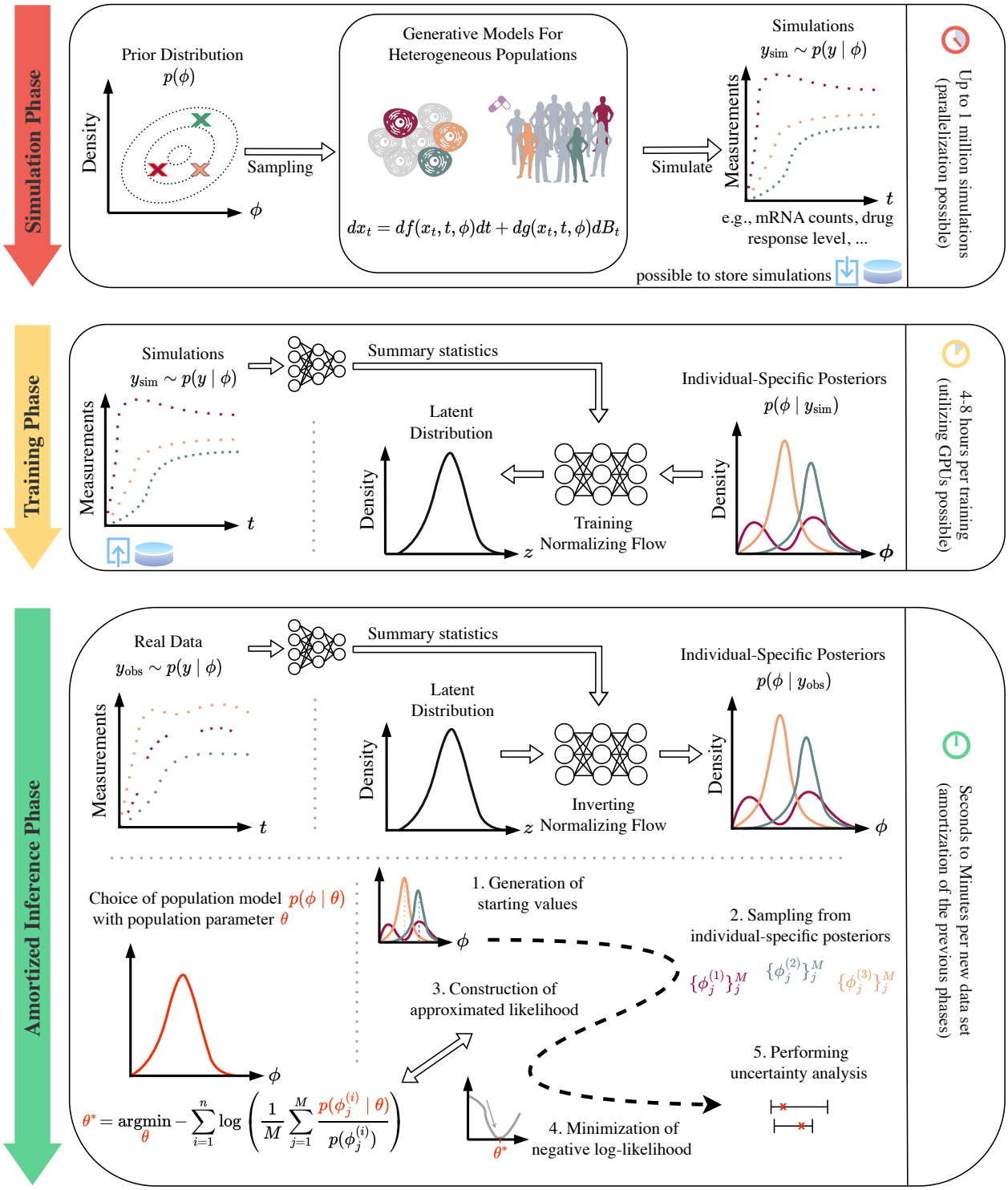


Figure 6: *Detailed concept visualization* of the neural posterior estimation based amortized approach to NLME model inference.

4.2 The non-linear mixed-effects model

Having a description of a specific individual in a population, we incorporate the heterogeneity of the population – as in the standard NLME frameworks [12] – by connecting the parameters from the individual level to the parameters describing the population. In NLME models, it is assumed that the population can be described by fixed effects $\boldsymbol{\beta}$, effects common to all individuals or certain groups of the population, and random effects specific to individuals $\boldsymbol{b}^{(i)}$ [12]. The random effects can then be described by a distribution, where we allow any valid probability distribution with a density, such as (log)-normal distributions, Cauchy distributions, and mixture distributions, among others. We relate these effects to individual-specific parameters $\boldsymbol{\phi}^{(i)}$ using a *population model* f, such that $\boldsymbol{\phi}^{(i)} = f(\boldsymbol{\beta}, \boldsymbol{b}^{(i)})$. Here, f is often a simple linear combination or an exponential and is an abstraction of the standard non-linear mixed-effects model in [12]. For ease of notation, we consider a single vector of population parameters $\boldsymbol{\theta}$ that fully characterize the distribution of individual-specific parameters $\boldsymbol{\phi}^{(i)}$. For example, this can be $\boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{D})$ with random effects $\boldsymbol{b}^{(i)} \sim \mathcal{N}(0, \boldsymbol{D})$ and $\boldsymbol{\phi}^{(i)} = A\boldsymbol{\beta} + B\boldsymbol{b}^{(i)}$, where A and B are design matrices, $\boldsymbol{\beta}$ are fixed effects, and $\boldsymbol{D}$ is the covariance matrix of random effects. Furthermore, the generative model $\mathcal{M}$ or the population model can include covariates $\boldsymbol{x}^{(i)}$, that is, additional information on individuals, for example, $\boldsymbol{\phi}^{(i)} = A\boldsymbol{\beta} + B\boldsymbol{b}^{(i)} + C\boldsymbol{x}^{(i)}$, where C is the design matrix for the covariates.

Our objective is to maximize the joint likelihood $p(\mathcal{D} \mid \boldsymbol{\theta})$ of the data $\mathcal{D}$ given the population parameters $\boldsymbol{\theta}$. This is a time-consuming task, as it involves repeated integration over unobserved random effects:

$$p(\mathcal{D} \mid \boldsymbol{\theta}) = \prod_{i=1}^{n} \int p(\boldsymbol{y}^{(i)} \mid \boldsymbol{\phi}) p(\boldsymbol{\phi} \mid \boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\phi}. \tag{4}$$

Solving this marginalization efficiently is the main challenge in parameter inference in non-linear mixed-effects models. Moreover, the conditional density $p(\boldsymbol{\phi} \mid \boldsymbol{\theta})$ is known from the population model specification, but the marginal likelihood $p(\boldsymbol{y}^{(i)} \mid \boldsymbol{\phi})$ could be intractable, for example, when the generative model is a stochastic differential equation.

4.3 Individual-specific neural posterior estimator

In the following, we develop an approach to efficiently maximize $p(\mathcal{D} \mid \boldsymbol{\theta})$ under the assumption that we can easily sample from an approximation of the posterior distribution $p(\boldsymbol{\phi} \mid \boldsymbol{y}^{(i)})$. In general, individual measurements are not sufficiently informative to obtain reliable point estimates and only the joint information is reliable [12]. However, using a Bayesian approach to describe individuals, we encode all the available information on a specific individual i in the posterior of the parameters $\boldsymbol{\phi}^{(i)}$ and then combine samples from the posterior to infer the population characteristics. Therefore, all parameters – also those which are considered constant in the population – will first be treated as random variables. For that, we consider the parameter prior distribution $p(\boldsymbol{\phi})$ and define the joint distribution of parameters and observables $p(\boldsymbol{\phi}, \boldsymbol{y}) = p(\boldsymbol{y} \mid \boldsymbol{\phi}) \cdot p(\boldsymbol{\phi}) = p(\boldsymbol{\phi} \mid \boldsymbol{y}) \cdot p(\boldsymbol{y})$ using Bayes theorem. By sampling from the prior distribution $\boldsymbol{\phi} \sim p(\boldsymbol{\phi})$ and generating simulations from $\mathcal{M}(\boldsymbol{\phi})$, which correspond to the marginalized likelihood $p(\boldsymbol{y} \mid \boldsymbol{\phi})$, we get pairs of parameters and data $(\boldsymbol{\phi}, \boldsymbol{y}) \sim p(\boldsymbol{y} \mid \boldsymbol{\phi}) \cdot p(\boldsymbol{\phi})$. We use these pairs to train a normalizing flow from a normal distribution conditioned on observations $\boldsymbol{y}$ to the posterior distribution $p(\boldsymbol{\phi} \mid \boldsymbol{y})$ by minimizing the Kullback-Leibler divergence between the true and approximate posterior distributions as in [27]:

$$\arg\min_{\boldsymbol{\psi}} \mathbb{E}_{p(\boldsymbol{y})} \Big[\mathrm{KL}(p(\boldsymbol{\phi} \mid \boldsymbol{y}) \mid\mid q_{\boldsymbol{\psi}}(\boldsymbol{\phi} \mid \boldsymbol{y})) \Big] = \arg\max_{\boldsymbol{\psi}} \iint p(\boldsymbol{y}, \boldsymbol{\phi}) \log q_{\boldsymbol{\psi}}(\boldsymbol{\phi} \mid \boldsymbol{y}) \, \mathrm{d}\boldsymbol{y} \, \mathrm{d}\boldsymbol{\phi}. \tag{5}$$

The approximation $q_{\boldsymbol{\psi}}(\boldsymbol{\phi} \mid \boldsymbol{y})$ of the posterior can be expressed by a density transformation from the latent normal distribution and, therefore, can be efficiently evaluated. By minimizing (5) using a Monte Carlo approximation, we train a global approximation of the posterior distribution $p(\boldsymbol{\phi} \mid \boldsymbol{y})$ for any parameters and data $(\boldsymbol{\phi}, \boldsymbol{y})$. In particular, we parameterize the normalizing flow by an invertible neural network and train it together with a summary network (Figure 6).

The summary network provides informative low-dimensional summary statistics on the observations and should be adapted to the problem at hand. For time trajectories, we use long-short-term memory neural networks to ensure that regardless of the number of observations we get a fixed length vector of summary statistics, which is important, as the invertible neural network have a fixed dimension. Besides the restriction of getting a fixed length vector as summary statistics, one can use any architecture as summary network, such as transformers, convolutional neural networks, etc., but also fixed summary statistics if the sufficient statistics are known. This allows us to work with a variety of different simulations, ranging from snapshot data to densely measured observations. If the covariates are part of the generative model $\mathcal{M}$ and not of the population model f, they must be simulated during the training phase of the neural posterior estimator in the same way as the other parameters of the generative model. Almost all the total computational cost is required to simulate the observations and train the neural networks. The simulation time, which can easily be larger than the training time, depends on the generative model and the number of simulations needed for training. It can be effectively reduced by heavy parallelization since all simulations are independent, whereas the training time depends on the number of simulations and parameters in the generative model. Training time is reduced by using GPUs and early stopping, where the latter is also assumed to improve the generalization of neural networks [50]. After training the normalizing flow, convergence can be ensured through calibration diagnostics (see Supplement A.3). Then, we are able to efficiently sample from the posterior distributions that are conditioned on individual-specific observations, which will allow us to estimate the distribution of the population.

4.4 Problem reformulation allows use of pre-trained density

Given individual-specific posterior distributions $p(\boldsymbol{\phi} \mid \boldsymbol{y}^{(i)})$, we can proceed to estimate population parameters $\boldsymbol{\theta}$ by reformulating the problem. We can rewrite the integrals over the marginal likelihood (4) as a conditional expectation

$$p(\mathcal{D} \mid \boldsymbol{\theta}) = \prod_{i=1}^{n} p(\boldsymbol{y}^{(i)}) \int p(\boldsymbol{\phi} \mid \boldsymbol{y}^{(i)}) \frac{p(\boldsymbol{\phi} \mid \boldsymbol{\theta})}{p(\boldsymbol{\phi})} \, \mathrm{d}\boldsymbol{\phi} = \prod_{i=1}^{n} p(\boldsymbol{y}^{(i)}) \mathbb{E}_{\boldsymbol{\phi} \sim p(\boldsymbol{\phi} \mid \boldsymbol{y}^{(i)})} \left[\frac{p(\boldsymbol{\phi} \mid \boldsymbol{\theta})}{p(\boldsymbol{\phi})} \right], \tag{6}$$

provided that the prior $p(\boldsymbol{\phi})$ is non-zero in the integration domain. Here, the prior has the role of importance weights.

We can sample from the approximate posterior $q_{\boldsymbol{\psi}}(\boldsymbol{\phi} \mid \boldsymbol{y}^{(i)})$ and approximate the conditional expectation by a Monte Carlo sample. Further using the log-likelihood, which is commonly done for numerical stability [30], we arrive at the minimization problem

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} - \log(p(\mathcal{D} \mid \boldsymbol{\theta})) \approx \arg\min_{\boldsymbol{\theta}} - \sum_{i=1}^{n} \log \left(\frac{1}{M} \sum_{j=1}^{M} \frac{p(\boldsymbol{\phi}_j^{(i)} \mid \boldsymbol{\theta})}{p(\boldsymbol{\phi}_j^{(i)})} \right) \tag{7}$$

with $\boldsymbol{\phi}_j^{(i)} \sim q_{\boldsymbol{\psi}}(\boldsymbol{\phi} \mid \boldsymbol{y}^{(i)})$ i.i.d. for $j = 1, \ldots, M$, for each individual i. This problem can be solved with a gradient-based optimizer; here we used the local optimization method L-BFGS [51]. The minimization is computationally efficient and simple, as no numerical simulations of the underlying model are required. Since we have independent samples, we do not need a large sample size M. Therefore, the computational costs of inferring population parameters are negligible.

4.5 Implementation

We implemented the individual-specific posterior approximation using the `BayesFlow` tool [52]. For a specification of the neural network architecture, we refer to the Supplement A.3. To estimate the population parameters, we implemented the minimization problem (7) as an objective function in the `pyPESTO` toolbox [33]. There, we used the local optimization method L-BFGS [51] embedded in a multistart framework with starting points calculated from the medians of the individual-specific posteriors (e.g., mean and covariance for a normal distribution). In our applications, usually 10 starts were already enough to reliably obtain the global optimum several times, but it is easy to perform more. Parameters that are shared between individuals, that is, parameters which do not consist of a random effect, can be approximated in the given approach by fixing their variance to a small value. Note that the objective (7) often reduces to a logarithmic sum of exponentials, for which numerically stable implementations should be used, such as the log-sum-exp-trick [53]. The simulations of the generative model, multistarts in `pyPESTO` and a single start in `Monolix` used all available cores for parallelization. Moreover, the contribution of each individual could also be evaluated in parallel, giving the option of further parallelizing the calculation of the objective in a single start.

We ran all analyses on a computing cluster using eight CPU cores for parallelization and one GPU for training the neural networks. The computing cluster uses an AMD EPYC 7F72 with a clock speed up to 3.2 GHz and 1 TB of RAM. The neural network training was performed on a cluster node with an NVIDIA A100 graphics card with 40 GB of VRAM.

The code and a guide, aimed at assisting users in training their own non-linear mixed-effects models, can be found at `https://github.com/arrjon/Amortized-NLME-Models.git`. A snapshot of the code and the results underlying this study can be found at `https://zenodo.org/record/8245786`. The patient data cannot be disclosed, while the single-cell data has been made available by Fröhlich *et al.* [25].

Acknowledgments

This work was supported by the German Federal Ministry of Education and Research (BMBF) (EMUNE/031L0293C and FitMultiCell/031L0159C), the German Research Foundation (DFG) under Germany's Excellence Strategy (EXC 2047 390873048 and EXC 2151 390685813 and the Schlegel Professorship for J.H.), and the EU Horizon 2020 program (ORCHESTRA; 101016167). Y.S. acknowledges financial support by the Joachim Herz Stiftung.

Author Contributions

J.A., Y.S., and J.H. conceptualized the amortized approach to NLME models. J.A. implemented and analyzed the method. C.P. performed the analysis of the single-cell models in `Monolix`. O.T. and U.J. performed the analysis of the pharmacokinetic model in `NONMEM`. J.A., Y.S., and J.H. wrote the manuscript. All authors discussed and approved the final manuscript.

Competing Interests

The authors declare no competing interests.

References

1. Liston, A., Humblet-Baron, S., Duffy, D. & Goris, A. Human immune diversity: from evolution to modernity. *Nature Immunology* **22,** 1479–1489. https://doi.org/10.1038/s41590-021-01058-1 (2021).
2. Brodin, P. & Davis, M. M. Human immune system variation. *Nat Rev Immunol* **17,** 21–29 (Jan. 2017).
3. Claret, L. *et al.* Model-based prediction of phase III overall survival in colorectal cancer on the basis of phase II tumor dynamics. *Journal of Clinical Oncology* **27,** 4103–4108 (2009).
4. Ribba, B. *et al.* A review of mixed-effects models of tumor growth and effects of anticancer drug treatment used in population analysis. *CPT: pharmacometrics & systems pharmacology* **3,** 1–10 (2014).
5. Groenland, S. L., Mathijssen, R. H., Beijnen, J. H., Huitema, A. D. & Steeghs, N. Individualized dosing of oral targeted therapies in oncology is crucial in the era of precision medicine. *European Journal of Clinical Pharmacology* **75,** 1309–1318 (2019).
6. Spencer, S. L., Gaudet, S., Albeck, J. G., Burke, J. M. & Sorger, P. K. Non-genetic origins of cell-to-cell variability in TRAIL-induced apoptosis. *Nat.* **459,** 428–433 (May 2009).
7. Swain, P. S., Elowitz, M. B. & Siggia, E. D. Intrinsic and extrinsic contributions to stochasticity in gene expression. *Proc. Natl. Acad. Sci. USA* **99,** 12795–12800 (Oct. 2002).
8. Goldstein, H. *Multilevel models in education and social research* (Oxford University Press, 1987).
9. Yu, Z. *et al.* Beyond t test and ANOVA: applications of mixed-effects models for more rigorous statistical analysis in neuroscience research. *Neuron* **110,** 21–35. https://doi.org/10.1016/j.neuron.2021.10.030 (2022).
10. Llamosi, A. *et al.* What Population Reveals about Individual Cell Identity: Single-cell Parameter Estimation of Models of Gene Expression in Yeast. *PLoS Comput. Biol* **12,** 1–18. https://doi.org/10.1371/journal.pcbi.1004706 (Feb. 2016).
11. Bortz, D. M. & Nelson, P. W. Model Selection and Mixed-Effects Modeling of HIV Infection Dynamics. *Bulletin of Mathematical Biology* **68,** 2005–2025. https://doi.org/10.1007/s11538-006-9084-x (2006).
12. Pinheiro, J. C. *Topics in mixed effects models* Ph.D. thesis (University of Wisconsin, Madison, Madison, USA, 1994).
13. Beal, S. & Sheiner, L. The NONMEM System. *Am. Stat.* **34,** 118–119. ISSN: 00031305. http://www.jstor.org/stable/2684123 (1980).
14. Lindstrom, M. & Bates, D. Nonlinear mixed effects models for repeated measures data, Biometrics. *Biometrics.* https://www.ncbi.nlm.nih.gov/pubmed/2242409 (Sept. 1990).
15. Sheiner, L. B. & Beal, S. L. Evaluation of methods for estimating population pharmacokinetic parameters. I. Michaelis-menten model: Routine clinical pharmacokinetic data. *Journal of Pharmacokinetics and Biopharmaceutics* **8,** 553–571. https://doi.org/10.1007/BF01060053 (1980).
16. Pinheiro, J. C. & Bates, D. M. Approximations to the Log-Likelihood Function in the Nonlinear Mixed-Effects Model. *Journal of Computational and Graphical Statistics* **4,** 12–35. ISSN: 1061-8600. JSTOR: 1390625. https://www.jstor.org/stable/1390625 (2023) (1995).

17. Wang, Y. Derivation of Various NONMEM Estimation Methods. *Journal of Pharmacokinetics and Pharmacodynamics* **35,** 249–249. ISSN: 1573-8744. https://doi.org/10.1007/s10928-008-9083-7 (2023) (Apr. 1, 2008).

18. Ge, Z., J. Bickel, P. & A. Rice, J. An Approximate Likelihood Approach to Nonlinear Mixed Effects Models via Spline Approximation. *Computational Statistics & Data Analysis* **46,** 747–776. ISSN: 0167-9473. https://www.sciencedirect.com/science/article/pii/S0167947303002330 (2023) (July 1, 2004).

19. Jönsson, S., Kjellsson, M. C. & Karlsson, M. O. Estimating Bias in Population Parameters for Some Models for Repeated Measures Ordinal Data Using NONMEM and NLMIXED. *Journal of Pharmacokinetics and Pharmacodynamics* **31,** 299–320. ISSN: 1573-8744. https://doi.org/10.1023/B:JOPA.0000042738.06821.61 (2023) (Aug. 1, 2004).

20. Kuhn, E. & Lavielle, M. Maximum likelihood estimation in nonlinear mixed effects models. *Comput. Stat. Data. Anal.* **49,** 1020–1038. ISSN: 0167-9473. http://www.sciencedirect.com/science/article/pii/S0167947304002221 (2005).

21. Lee, S. Y. Bayesian Nonlinear Models for Repeated Measurement Data: An Overview, Implementation, and Applications. *Mathematics* **10,** 898. ISSN: 2227-7390. https://www.mdpi.com/2227-7390/10/6/898 (2023) (Mar. 11, 2022).

22. Augustin, D. *et al.* Filter Inference: A Scalable Nonlinear Mixed Effects Inference Approach for Snapshot Time Series Data. *bioRxiv* (2022).

23. Wiqvist, S., Golightly, A., McLean, A. T. & Picchini, U. Efficient Inference for Stochastic Differential Equation Mixed-Effects Models Using Correlated Particle Pseudo-Marginal Algorithms. *Computational Statistics & Data Analysis* **157,** 107151. ISSN: 0167-9473. https://www.sciencedirect.com/science/article/pii/S0167947320302425 (2023) (May 1, 2021).

24. Botha, I., Kohn, R. & Drovandi, C. Particle Methods for Stochastic Differential Equation Mixed Effects Models. *Bayesian Analysis* **16,** 575–609. https://doi.org/10.1214/20-BA1216 (2021).

25. Fröhlich, F. *et al.* Multi-experiment nonlinear mixed effect modeling of single-cell translation kinetics after transfection. *npj Systems Biology and Applications* **5,** 1. https://doi.org/10.1038/s41540-018-0079-7 (2018).

26. Persson, S. *et al.* Scalable and flexible inference framework for stochastic dynamic single-cell models. *PLoS Computational Biology* **18,** e1010082 (2022).

27. Radev, S. T., Mertens, U. K., Voss, A., Ardizzone, L. & Köthe, U. BayesFlow: Learning complex stochastic models with invertible neural networks. *IEEE transactions on neural networks and learning systems* (2020).

28. Cranmer, K., Brehmer, J. & Louppe, G. The Frontier of Simulation-Based Inference. *Proceedings of the National Academy of Sciences* **117,** 30055–30062 (2020).

29. Lixoft SAS, a. S. P. c. *Monolix 2023R1* 2023.

30. Raue, A. *et al.* Lessons learned from quantitative dynamical modeling in systems biology. *PLoS ONE* **8,** e74335 (Sept. 2013).

31. Pardi, N., Hogan, M. J., Porter, F. W. & Weissman, D. mRNA vaccines – a new era in vaccinology. *Nature Reviews Drug Discovery* **17,** 261–279. https://doi.org/10.1038/nrd.2017.243 (2018).

32. Talts, S., Betancourt, M., Simpson, D., Vehtari, A. & Gelman, A. Validating Bayesian Inference Algorithms with Simulation-Based Calibration. arXiv: 1804.06788 [stat]. http://arxiv.org/abs/1804.06788 (2022) (Oct. 21, 2020). preprint.

33. Schälte, Y. *et al. pyPESTO: A modular and scalable tool for parameter estimation for dynamic models* 2023.

34. Maier, C., Hartung, N., de Wiljes, J., Kloft, C. & Huisinga, W. Bayesian data assimilation to support informed decision making in individualized chemotherapy. *CPT: pharmacometrics & systems pharmacology* **9,** 153–164 (2020).

35. Fröhlich, F., Theis, F. J. & Hasenauer, J. *Uncertainty analysis for non-identifiable dynamical systems: Profile likelihoods, bootstrapping and more* in *Proc. 12th Int. Conf. Comp. Meth. Syst. Biol.* (eds Mendes, P., Dada, J. O. & Smallbone, K. O.) (Springer International Publishing Switzerland, Nov. 2014), 61–72.

36. Kreutz, C., Raue, A., Kaschek, D. & Timmer, J. Profile likelihood in systems biology. *FEBS J* **280,** 2564–2571 (2013).

37. Wilkinson, D. J. Stochastic modelling for quantitative description of heterogeneous biological systems. *Nat. Rev. Genet.* **10,** 122–133 (Feb. 2009).

38. Stumpf, P. S. *et al.* Stem cell differentiation as a non-Markov stochastic process. *Cell Systems* **5,** 268–282 (2017).

39. Zechner, C., Unger, M., Pelet, S., Peter, M. & Koeppl, H. Scalable inference of heterogeneous reaction kinetics from pooled single-cell recordings. *Nat. Methods* **11,** 197–202 (Jan. 2014).

40. Sisson, S. A., Fan, Y. & Beaumont, M. *Handbook of approximate Bayesian computation* (Chapman and Hall/CRC, 2018).

41. Pieschner, S., Hasenauer, J. & Fuchs, C. Identifiability analysis for models of the translation kinetics after mRNA transfection. *bioRxiv.* eprint: https://www.biorxiv.org/content/early/2021/05/18/2021.05.18.444633.full.pdf. https://www.biorxiv.org/content/early/2021/05/18/2021.05.18.444633 (2021).

42. Diekstra, M. *et al.* Population Modeling Integrating Pharmacokinetics, Pharmacodynamics, Pharmacogenetics, and Clinical Outcome in Patients With Sunitinib-Treated Cancer. *CPT: Pharmacometrics & Systems Pharmacology* **6,** 604–613. ISSN: 2163-8306. https://onlinelibrary.wiley.com/doi/abs/10.1002/psp4.12210 (2023) (2017).

43. Savic, R. M. & Karlsson, M. O. Importance of Shrinkage in Empirical Bayes Estimates for Diagnostics: Problems and Solutions. *The AAPS Journal* **11,** 558–569. ISSN: 1550-7416. https://doi.org/10.1208/s12248-009-9133-0 (2023) (Sept. 1, 2009).

44. Gillespie, D. T. Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.* **81,** 2340–2361 (Dec. 1977).

45. Schmitt, M., Bürkner, P.-C., Köthe, U. & Radev, S. T. *Detecting Model Misspecification in Amortized Bayesian Inference with Neural Networks* arXiv: arXiv:2112.08866. http://arxiv.org/abs/2112.08866 (2023). preprint.

46. Hermans, J. *et al. A Trust Crisis In Simulation-Based Inference? Your Posterior Approximations Can Be Unfaithful* arXiv: 2110.06581 [cs, stat]. http://arxiv.org/abs/2110.06581 (2023). preprint.

47. Hagemann, P. L., Hertrich, J. & Steidl, G. *Generalized Normalizing Flows via Markov Chains* in (Cambridge University Press, Feb. 28, 2023). https://www.cambridge.org/core/product/identifier/9781009331012/type/element (2023).

48. Kingma, D. P. & Welling, M. *Auto-Encoding Variational Bayes* arXiv: 1312.6114 [cs, stat]. http://arxiv.org/abs/1312.6114 (2023). preprint.

49. Wang, Y. & Ročková, V. *Adversarial Bayesian Simulation* arXiv: 2208.12113 [stat]. http://arxiv.org/abs/2208.12113 (2023). preprint.

50. Zhang, C., Bengio, S., Hardt, M., Recht, B. & Vinyals, O. Understanding Deep Learning (Still) Requires Rethinking Generalization. *Commun. ACM* **64,** 107–115. https://doi.org/10.1145/3446776 (Jan. 2021).

51. Liu, D. C. & Nocedal, J. On the limited memory BFGS method for large scale optimization. *Math. Program.* **45,** 503–528 (1989).

52. Radev, S. T. *et al. BayesFlow: Amortized Bayesian Workflows With Neural Networks* 2023.

53. Blanchard, P., Higham, D. J. & Higham, N. J. Accurately computing the log-sum-exp and softmax functions. *IMA Journal of Numerical Analysis* **41,** 2311–2330 (2021).

54. Rackauckas, C. & Nie, Q. Differentialequations.jl – a performant and feature-rich ecosystem for solving differential equations in julia. *Journal of open research software* **5** (2017).

55. Lam, S. K., Pitrou, A. & Seibert, S. *Numba: A LLVM-Based Python JIT Compiler* in *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC* (Association for Computing Machinery, Austin, Texas, 2015). ISBN: 9781450340052. https://doi.org/10.1145/2833157.2833162.

56. Yu, H. *et al.* Integrated semi-physiological pharmacokinetic model for both sunitinib and its active metabolite SU 12662. *British Journal of Clinical Pharmacology* **79,** 809–819 (2015).

A Supplementary Information

A.1 Specification of the single-cell models

Living cells exhibit variability at the single cell level due to various factors such as cellular processes, cell cycle state, environmental differences, and individual cell history [25]. Fröhlich *et al.* were interested in the dynamics of protein expression and transfected single cells with enhanced green fluorescent protein (eGFP) encoding mRNA. The expression of the eGFP reporter gene was recorded every ten minutes for a period of 30 hours using a scanning time-lapse microscope setup. From these data, the authors estimated the parameters of the translation process using ordinary differential equation (ODE) models in a NLME framework.

In this work, we focus on two models termed the "simple" and "detailed" models from [25]. We denote the abundance of mRNA as m, proteins as p, ribosomes as r, and enzymes as e. For both models, we assume additive normal measurement noise, that is, the measurements y follow $y \sim N(0, \sigma^2)$ and our assumed prior distribution for σ is $\log N(-1, 2)$.

Simple ODE model

The ODE system is

$$\begin{aligned} \frac{dm}{dt} &= -\delta \cdot m & m(t_0) &= 1 \\ \frac{dp}{dt} &= k \cdot m_0 \cdot \text{scale} \cdot m - \gamma \cdot p & p(0) &= 0 \\ y &= \log(p + \text{offset}), \end{aligned}$$

where the priors assumed for the variables are

- mRNA degradation rate $\delta \sim \log \mathcal{N}(-3, 5)$,
- protein degradation rate $\gamma \sim \log \mathcal{N}(-3, 5)$,
- combined parameters $k \cdot m_0 \cdot \text{scale} \sim \log \mathcal{N}(5, 11)$ (referred to as scale),
- mRNA entering the cell time point $t_0 \sim \log \mathcal{N}(0, 2)$,
- and offset $\sim \log \mathcal{N}(1, 6)$.

The parameters k, m_0, scale can only be identified as a product to improve identifiability [25]. This ODE system has an analytical solution, which we use to perform simulations in Python.

Detailed ODE model

The ODE system is

$$\begin{aligned} \frac{dm}{dt} &= -\delta_1 m_0 \cdot m \cdot e - k_1 m_0 \cdot m \cdot r + k_2 \cdot \left(\frac{r_0}{m_o} - r\right) & m(t_0) &= 1 \\ \frac{de}{dt} &= \delta_1 m_0 \cdot m \cdot e - \delta_2 \cdot \left(\frac{e_0}{m_0} - e\right) & e(0) &= \frac{e_0}{m_0} \\ \frac{dr}{dt} &= k_2 \cdot \left(\frac{r_0}{m_0} - r\right) - k_1 m_0 \cdot m \cdot r & r(0) &= \frac{r_0}{m_0} \\ \frac{dp}{dt} &= k_2 m_0 \text{scale} \cdot \left(\frac{r_0}{m_0} - r\right) - \gamma \cdot p & p(0) &= 0 \\ y &= \log(p + \text{offset}). \end{aligned}$$

For a detailed description of the parameters, we refer to [25]. The priors assumed for the variables are

$$\begin{aligned} \delta_1 m_0 &\sim \log\mathcal{N}(0,5), & k_1 m_0 &\sim \log\mathcal{N}(1,2) \\ \delta_2 &\sim \log\mathcal{N}(-1,5), & \frac{r_0}{m_0} &\sim \log\mathcal{N}(-3,2), \\ \frac{e_0}{m_0} &\sim \log\mathcal{N}(0,2), & \gamma &\sim \log\mathcal{N}(-6,5), \\ k_2 m_0 \text{scale} &\sim \log\mathcal{N}(12,1), & t_0 &\sim \log\mathcal{N}(0,2), \\ k_2 &\sim \log\mathcal{N}(0,2), & \text{offset} &\sim \log\mathcal{N}(2,6). \end{aligned}$$

To the combined parameters $k_2 m_0 \text{scale}$ we refer to as scale. This ODE system is simulated using the Rodas5P solver implemented in the Julia package `DifferentialEquations.jl` [54].

SDE model

The simple ODE model can be easily extended to the SDE model

$$d\begin{pmatrix} m \\ p \end{pmatrix} t = \begin{pmatrix} -\delta \cdot m(t) \\ k \cdot m(t) - \gamma \cdot p(t) \end{pmatrix} dt + \begin{pmatrix} \sqrt{\delta m(t)} & 0 \\ 0 & \sqrt{k \cdot m(t) + \gamma \cdot p(t)} \end{pmatrix} dB_t$$

from [41], where B_t is a two-dimensional standard Brownian motion, $m(t_0) = 1$ and $p(0) = 0$. To compare the model to the previous one we take as observable mapping

$$y = \log(\text{scale} \cdot p + \text{offset}).$$

The priors assumed for the variables are

$$\begin{aligned} \delta &\sim \log\mathcal{N}(-3,5), & \text{scale} &\sim \log\mathcal{N}(0,5), \\ \gamma &\sim \log\mathcal{N}(-3,5), & t_0 &\sim \log\mathcal{N}(0,2), \\ k &\sim \log\mathcal{N}(-1,5), & \text{offset} &\sim \log\mathcal{N}(1,5). \\ m_0 &\sim \log\mathcal{N}(5,5), \end{aligned}$$

This SDE system is simulated based on a Euler-Maruyama scheme with a step size of 0.01 and using just in time compilation from `numba` [55].

A.1.1 Multi-experiment setup

To increase parameter identifiability Fröhlich *et al.* introduced in [25] an experimental setup with two distinct variants of eGFP that differ in their protein lifetime (here referred to as eGFP and d2eGFP). The modeling assumption was that the two variants share all the parameters in the NLME models, but the protein degradation rate γ. Thus, we have to distinct data sets $\mathcal{D}_{\text{eGFP}}$ and $\mathcal{D}_{\text{eGFP}}$ for the two variants, respectively, and shared parameters $\boldsymbol{\theta}$, which leads to

$$p(\mathcal{D} \mid \tilde{\boldsymbol{\theta}}) = p(\mathcal{D}_{\text{eGFP}} \mid \boldsymbol{\theta}, \gamma_{\text{eGFP}}) + p(\mathcal{D}_{\text{d2eGFP}} \mid \boldsymbol{\theta}, \gamma_{\text{d2eGFP}}).$$

For our amortized approach, we could reuse the trained neural posterior estimator and only needed to change the population model with respect to the shared parameters

$$\boldsymbol{\phi}_{\gamma}^{(i)} = \mathbb{1}_{\boldsymbol{y}^{(i)} \in \mathcal{D}_{\text{eGFP}}} \cdot \gamma_{\text{eGFP}} + \mathbb{1}_{\boldsymbol{y}^{(i)} \in \mathcal{D}_{\text{d2eGFP}}} \cdot \gamma_{\text{d2eGFP}},$$

where the other entries of $\boldsymbol{\phi}^{(i)}$ are equal to the remaining mean parameters in $\boldsymbol{\theta}$.

A.1.2 Synthetic data

The synthetic data set is generated by setting the population parameters to reasonable values based on the results in [25] (see Table A1, A2 and A3) and then sampling random effects from a log-normal distribution until the desired number of synthetic cells is generated. Since we know all cell-specific parameters, we can compute the sample mean and covariance of the parameters, which are the optimal values that we would like to recover.

Table A1: Population parameters of log-normal distribution for synthetic data of simple single-cell NLME model.

parameter	δ	γ	$k \cdot m_0 \cdot \text{scale}$	t_0	offset	σ
mean	-0.694	-7.014	6.217	-0.164	2.079	-3.454
variance	0.941	7.014	0.004	0.116	0	0

Table A2: Population parameters of log-normal distribution for synthetic data of detailed single-cell NLME model.

parameter	$\delta_1 m_0$	δ_2	$e_0 m_0$	k_2-m_0-scale	k_2	$k_1 m_0$	$r_0 m_0$	γ	t_0	offset	σ
mean	-0.10144	-0.88443	-0.42549	13.81551	0.42143	0.97477	-3.50153	-6.91273	-0.34573	2.07944	-3.45388
variance	0.56752	0.74721	0.52594	0	0.44084	1.45996	2.3979	4.61512	0.48075	0	0

Table A3: Population parameters of log-normal distribution for synthetic data of the SDE single-cell NLME model.

parameter	δ	γ	k	m_0	scale	t_0	offset	σ
mean	-0.694	-7.014	0.027	5.704	0.751	-0.164	2.079	-3.454
variance	0.941	7.014	0.675	$6 \cdot 10^{-5}$	0	0.116	0	0

A.1.3 Further analysis of the single-cell NLME models

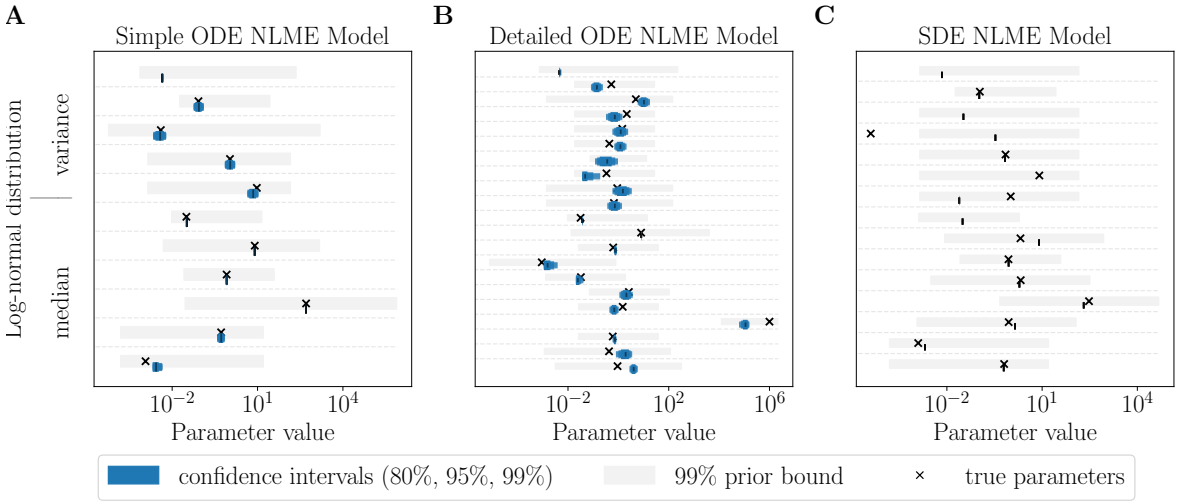


Figure A1: *Confidence intervals for the single-cell models on synthetic data.* Data was generated by (**A**) the simple ODE model, (**B**) the detailed ODE model and, (**C**) the SDE model. Then parameters and CIs (based on profile likelihoods) were estimated using the amortized approach to NLME models. True parameters which are 0, are not shown.

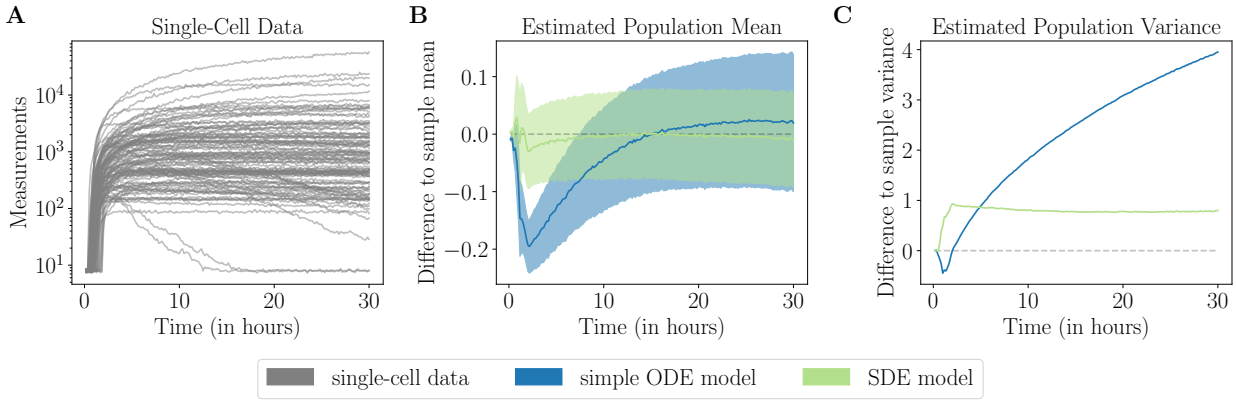


Figure A2: *Fit for SDE NLME model on synthetic data.* (**A**) Synthetic data describing single-cell translation kinetics after mRNA transfection generated by the SDE NLME model. (**B**–**C**) Difference of estimated population mean (**B**) and variance (**C**) over time of the SDE and ODE NLME model on synthetic data generated by the SDE model.

A.2 Specification of the pharmacokinetic model

Over the past two decades, many oral targeted therapies have been developed in the field of oncology, many of which target the angiogenesis of neoplasms, which plays an important role in tumor growth. However, angiogenesis inhibitors generally show high variability between patients, leading to significant differences in exposure [5]. Therefore, pharmacokinetic (PK) modeling is required to develop targeted dosing strategies for sub-populations or even in a personalized manner, discover concentration thresholds for toxicity, investigate potential interactions, and guide study planning, among other purposes. Sunitinib, an angiogenesis inhibitor, which belongs to the class of tyrosine kinase inhibitors, was the subject of the population pharmacokinetic model, which is described in more detail below: In the model developed by Diekstra *et al.* [42], the distribution of sunitinib is described by a single compartment model, while for its metabolite SU12662, a two compartment model was used. Presystemic metabolization was described according to the model by Yu *et al.* [56] by a hypothetical enzyme compartment. The hypothetical compartment was parameterized as follows, with Q_H being the calculated concentration:

$$\text{CLIV} = \frac{k_a \cdot A_D + Q_H \cdot \frac{A_{c,\text{sunitinib}}}{V_{c,\text{sunitinib}}}}{Q_H + CL_{\text{sunitinib}}}.$$

k_a denotes for the absorption rate constant while A_D and $A_{c,\text{sunitinib}}$ represent the amounts in the dosing or central compartment, respectively. $CL_{\text{sunitinib}}$ and $V_{c,\text{sunitinib}}$ denote the clearance and volume of distribution of the central compartment of sunitinib in this equation.

The model includes the sex and weight of the patients as covariates. Each patient i received a personal medication (DOS_i) and was measured over a different period of time and at varying time points. In the following, we present the model for each individual; therefore, the index i is removed. The patient's weight is normalized as follows

$$\text{wt} := \begin{cases} 83 & \text{if weight is missing and sex} = 1 \\ 75 & \text{if weight is missing and sex} = 0 \\ \text{weight} & \text{else} \end{cases}, \qquad \text{ASCL} := \left(\frac{\text{wt}}{70}\right)^{0.75}, \qquad \text{ASV} := \frac{\text{wt}}{70}.$$

The parameters we want to estimate are $\theta \in \mathbb{R}^{11}_{\geq 0}$, and $\eta \in \mathbb{R}^{4}_{\geq 0}$, which are incorporated in the ODE model as follows:

$$\begin{aligned} k_a &= \theta_1 \\ V_2 = V_{c,\text{sunitinib}} &= \theta_2 \cdot \text{ASV} \cdot \eta_1 \\ Q_H &= \theta_3 \cdot \text{ASCL} \\ \text{CL}_{\text{sunitinib}} &= \theta_4 \cdot \text{ASCL} \cdot \eta_3 \\ \text{CL}_{\text{SU12662}} &= \theta_5 \cdot \text{ASCL} \\ V_3 = V_{c,\text{SU12662}} &= \theta_6 \cdot \text{ASV} \cdot \eta_2 \end{aligned} \qquad \begin{aligned} Q_{34} &= \theta_7 \cdot \text{ASCL} \\ V_4 = V_{p,\text{SU12662}} &= \theta_8 \cdot \text{ASV} \\ f_m &= \theta_9 \cdot \eta_4 \\ Q_{25} &= \theta_{10} \cdot \text{ASCL} \\ V_5 = V_{p,\text{sunitinib}} &= \theta_{11} \cdot \text{ASV} \end{aligned}$$

and

$$
\begin{aligned}
\frac{dA_D}{dt} &= \frac{dA_1}{dt} = -k_a A_1 && A_1(0) = \text{DOS} \\
\frac{dA_{c,\text{sunitinib}}}{dt} &= \frac{dA_2}{dt} = Q_H \cdot \text{CLIV} - \frac{Q_H}{V_2} A_2 - \frac{Q_{25}}{V_2} A_2 + \frac{Q_{25}}{V_5} A_5 && A_2(0) = 0 \\
\frac{dA_{c,\text{SU12662}}}{dt} &= \frac{dA_3}{dt} = f_m \cdot \text{CL}_{\text{sunitinib}} \cdot \text{CLIV} - \frac{\text{CLM}}{V_3} A_3 - \frac{Q_{34}}{V_3} A_3 + \frac{Q_{34}}{V_4} A_4 && A_3(0) = 0 \\
\frac{dA_{p,\text{SU12662}}}{dt} &= \frac{dA_4}{dt} = \frac{Q_{34}}{V_3} A_3 - \frac{Q_{34}}{V_4} A_4 && A_4(0) = 0 \\
\frac{dA_{p,\text{sunitinib}}}{dt} &= \frac{dA_5}{dt} = \frac{Q_{25}}{V_2} A_2 - \frac{Q_{25}}{V_5} A_5 && A_5(0) = 0.
\end{aligned}
$$

As in the baseline [42], we fix $\theta_3 = 80$, $\theta_9 = 0.21$, and $\theta_{11} = 588$ to get comparable results.

Furthermore, whenever a patient takes medication (at t_j^{DOS}), we have

$$A_1(t_j^{\text{DOS}}) = \lim_{t \to t_j^{\text{DOS}}} A_1(t) + \text{DOS}.$$

As measurement function we apply

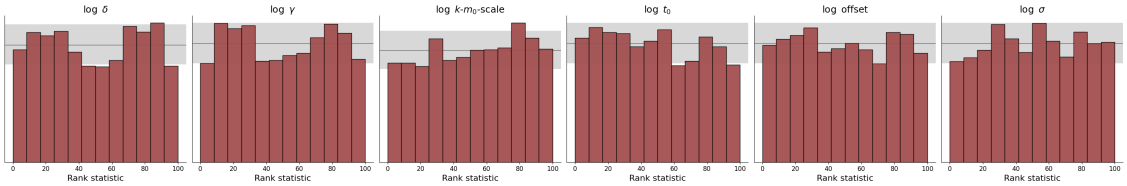
$$y_{2,3} = \theta_{12,13} \cdot \epsilon_{1,2} + \begin{cases} 0.001 & \text{if } A_{2,3} < 0.001 \\ \log(A_{2,3}) & \text{else,} \end{cases}$$

where $\epsilon_{2,3} \sim \mathcal{N}(0, \sigma^2)$. In [42], $\sigma^2 = 1$ was also fixed. This ODE system is simulated using the Rodas5P solver implemented in the Julia package `DifferentialEquations.jl` [54].

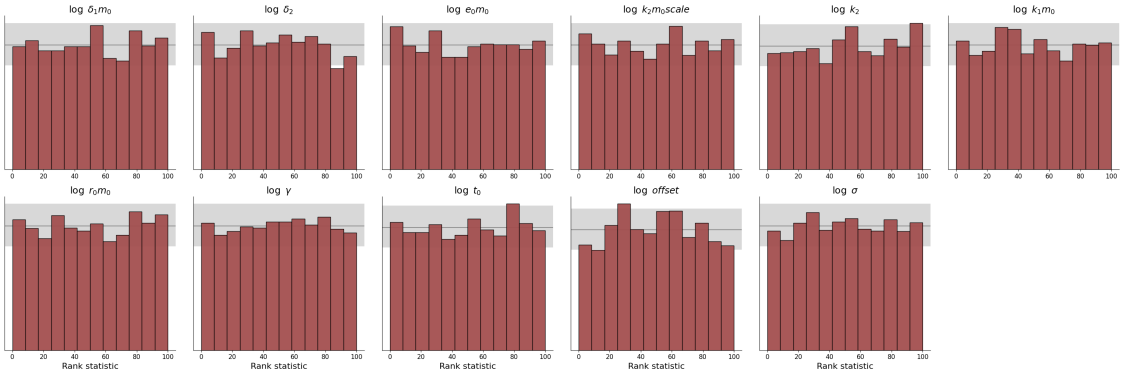
A.3 Calibration of the neural posterior estimator

To train the neural posterior estimator, we use `BayesFlow`, a flexible workflow to estimate normalizing flows with invertible neural networks [27]. Since all models describe trajectories over time, we chose a long short-term memory (LSTM) network with 2^d units as the basis of our summary network with d such that the number of units is larger than the number of observations given by the model and stacked coupling layers as an invertible neural network. For every model, multiple neural posterior estimators were trained. We varied the number of coupling layers from 6 to 8, added a $1d$-convolutional layer on top of the LSTMs and a dense layer at the end. Training consists of several epochs, and in each we generated 1000 batches of 128 simulations. Simulations can be either generated before or during training. Depending on the simulation time of the model, pre-simulation or online training is more efficient. We used online training for the simple ODE model, while we generated simulation beforehand for the other models. Training was stopped earlier if the loss calculated on a validation set did not improve any more. For the simple ODE model, we set a maximum of 500 epochs and, for all other models, a maximum of 1000 epochs. The error calculated on a validation set during training suggested convergence for all models (Supplement Figure A3). Furthermore, we checked the convergence of the neural posterior estimators based on their calibration plots, a diagnostic tool that comes with `BayesFlow`. Simulation-based calibration is a method to detect systematic biases in any Bayesian posterior sampling method [32]. Incorrect calibration can be seen by deviations from uniformity. All our estimators show no systematic bias (Supplement Figure A4, A6). Furthermore, for the best estimators, we assessed the validity of the individual-specific posteriors of the real data by comparing them with the posterior approximations given by an MCMC approximation with adaptive parallel tempering implemented in `PyPesto` [33]. In particular, the bimodal distributions of the parameters δ and γ in the simple ODE model are nicely recovered (Supplement Figure A5).

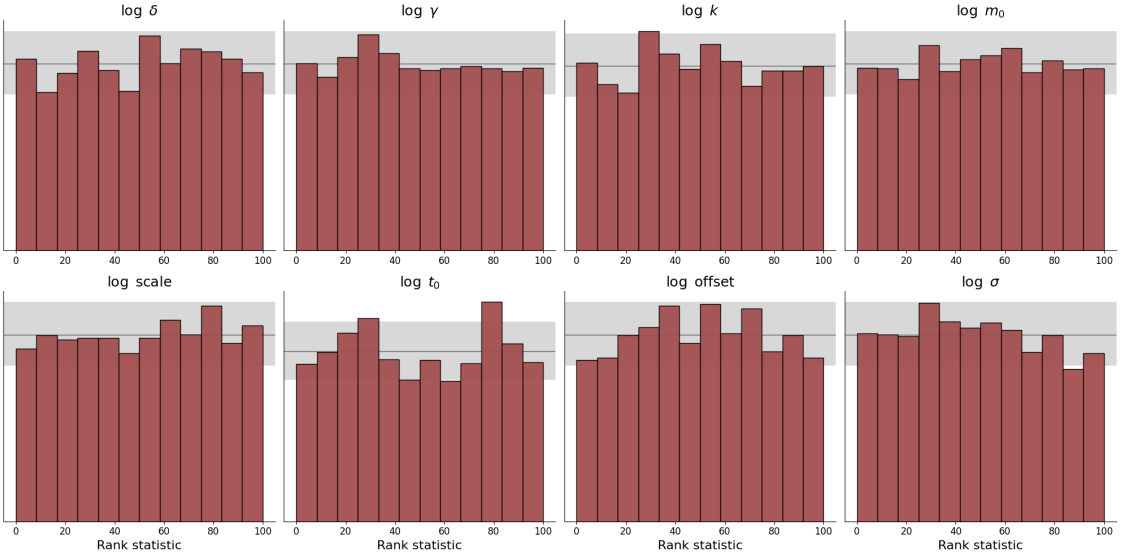
Figure A3: Exemplary loss during training of the simple ODE model.



(a) Simple ODE model

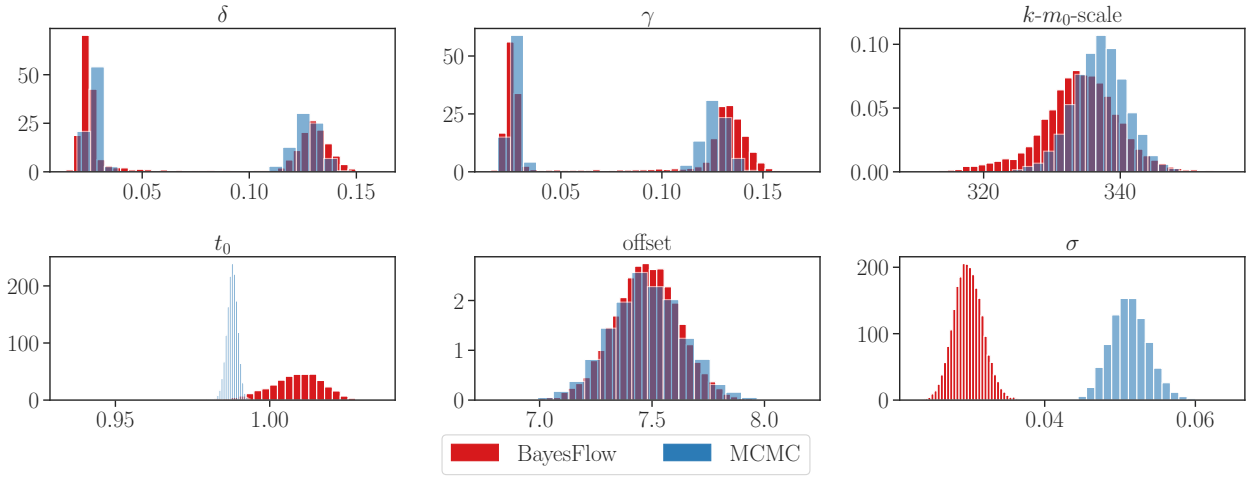


(b) Detailed ODE model

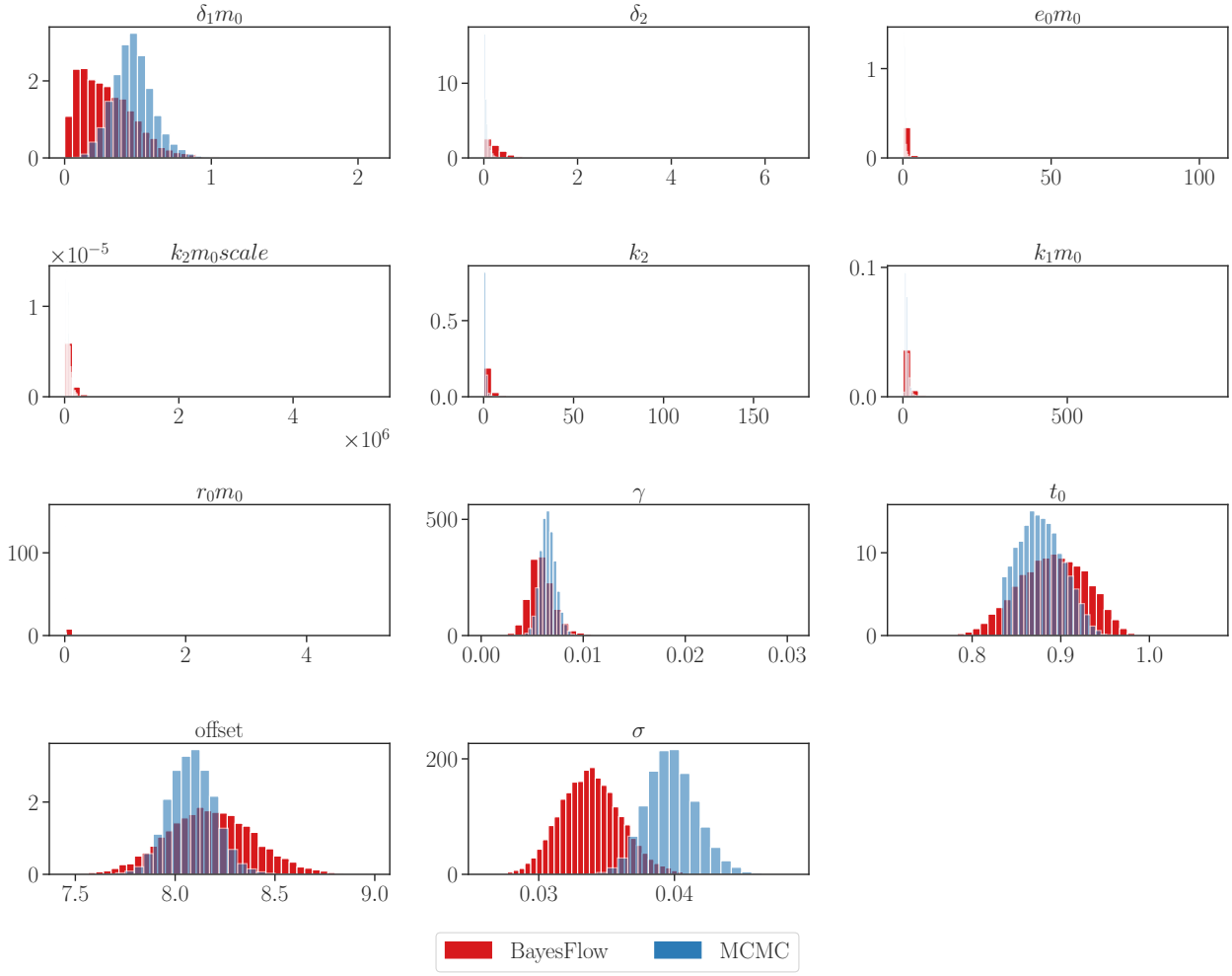


(c) SDE model

Figure A4: Simulation-based calibration plots of the individual posteriors for the (a) simple ODE, (b) detailed ODE and (c) SDE models. Incorrect calibration can be seen by deviations from uniformity (bars outside the gray area).



(a) Simple ODE model



(b) Detailed ODE model

Figure A5: Comparing individual-specific posteriors from a MCMC approximation and the neural posterior estimator for a single real cell in the (a) simple and the (b) detailed ODE model.

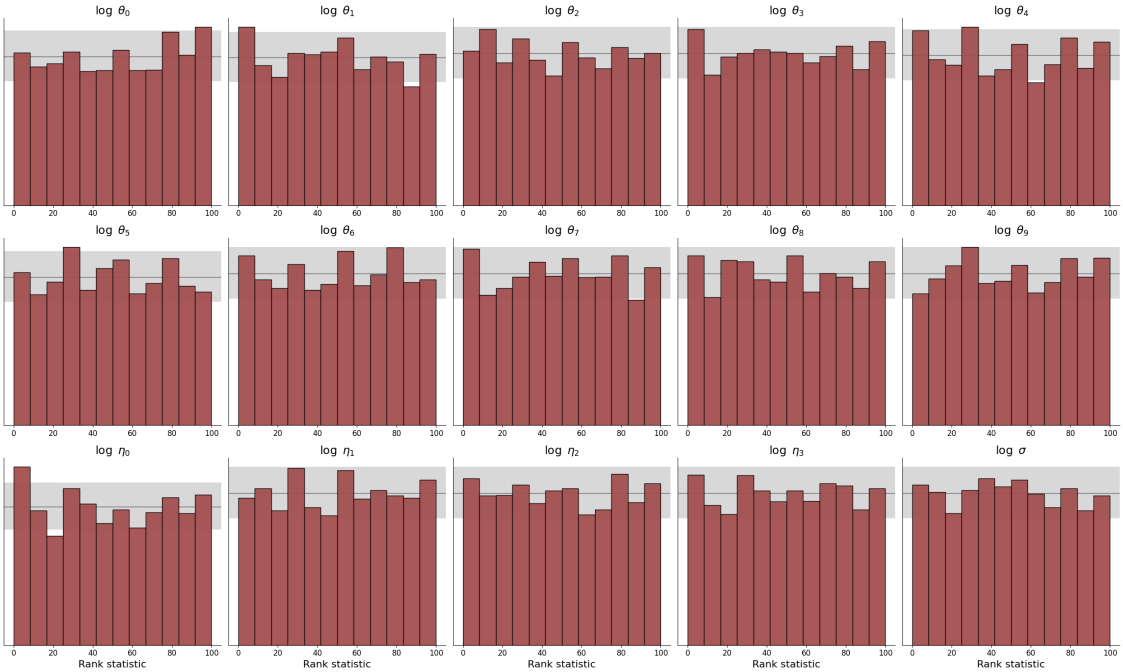


Figure A6: Calibration plot of the pharmacokinetic model.